INTRODUCTION TO SOFT COMPUTING

Prof. Debasis Samanta
Computer Science and Engineering
IIT Kharagpur

INDEX

Week 6

Week 7

Week 8

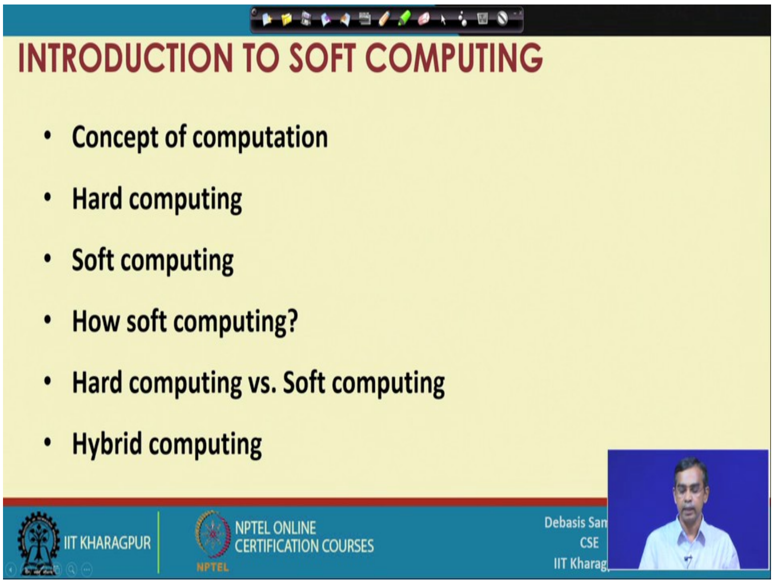
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 01

Introduction to soft computing

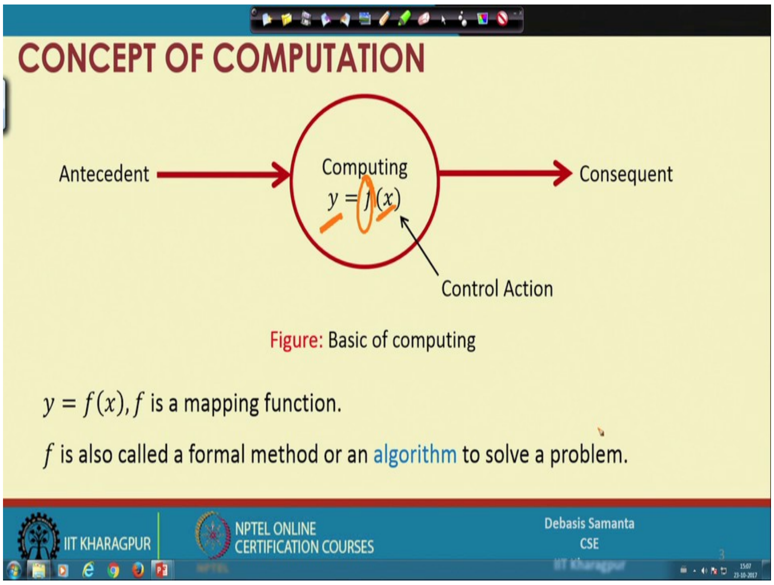
I take this opportunity to welcome you to the course Soft Computing. In today's lecture we will discuss about basic concept of soft computing. So, basically we know exactly soft computing is related to in some sets computing.

(Refer Slide Time: 00:25)



So, basically what is the concept of computation, we will learn about it. After these things as the term soft computing so there may be something which is called the hard computing so we learn about the hard computing next. And then in what way a soft computing is different from the hard computing. And then obviously, the natural question that arises that how the soft computing can be achieved and to understand better the soft computing we should know exactly what are the differences between the hard computing and soft computing. And there is also another concept it is basically the combination of the two computing paradigms, hard computing and soft computing it is called the hybrid computing. So, in today's lecture we will try to cover these are the different concepts here.

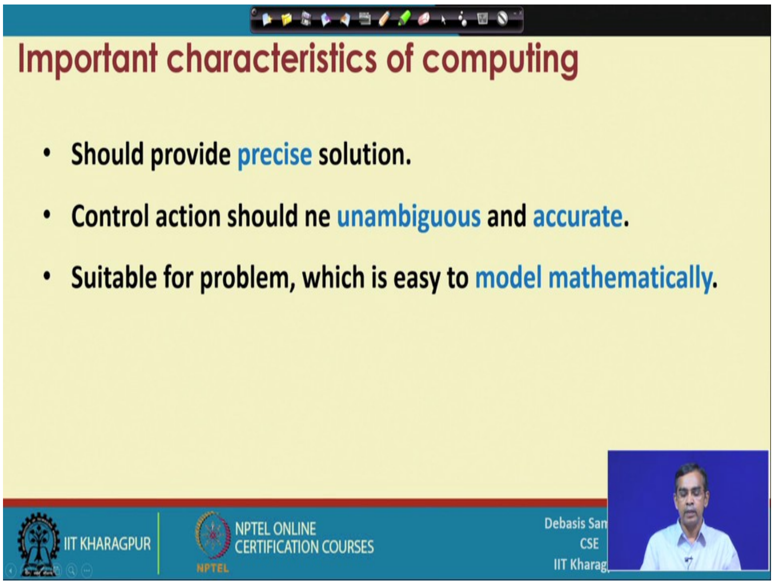
(Refer Slide Time: 01:24)



Now, let us first concept of computation. So, we know exactly a computing means there are certain input and then there is a procedure by which the input can be converted into some output. So, in the context of computing, so the input is called the antecedent and then output is called the consequence and then computing is basically mapping. Here we see, so f is the function f is the function basically which is responsible to convert x the input and to some output. So, this is the concept of computing is basically.

So, in other words, computing is nothing but is a mapping function, mapping from set of input to output. Now, this mapping is also alternatively called as formal method also it is called an algorithm, so basically algorithm to solve a problem.

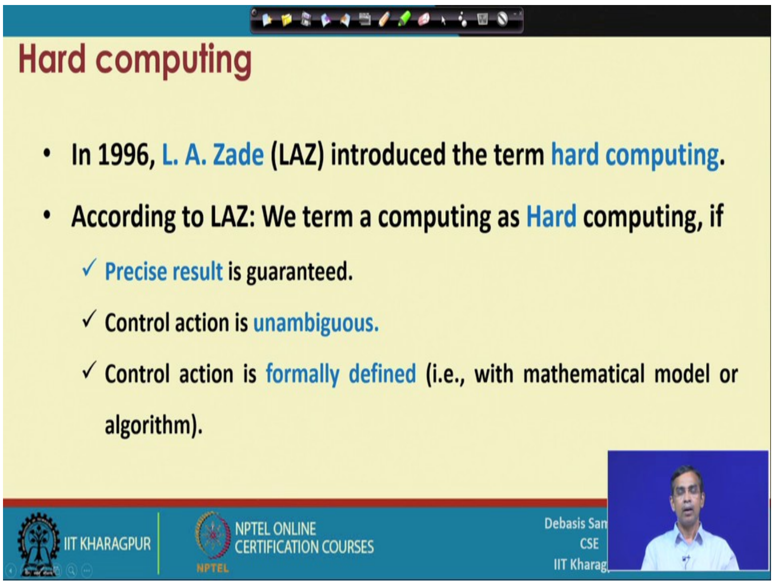
(Refer Slide Time: 02:36)



Now, say let us see, exactly what are the different characteristics of computing? So, computing is that, for a given input it always give a particular output. This means that it should provide a precise solution. And in order to achieve from a given set of input to an output it should follow some setup unambiguous and accurate step.

And the next characteristic is that, it should it is suitable for some problems which is easy to model mathematically. This means that for which there is an algorithm is available.

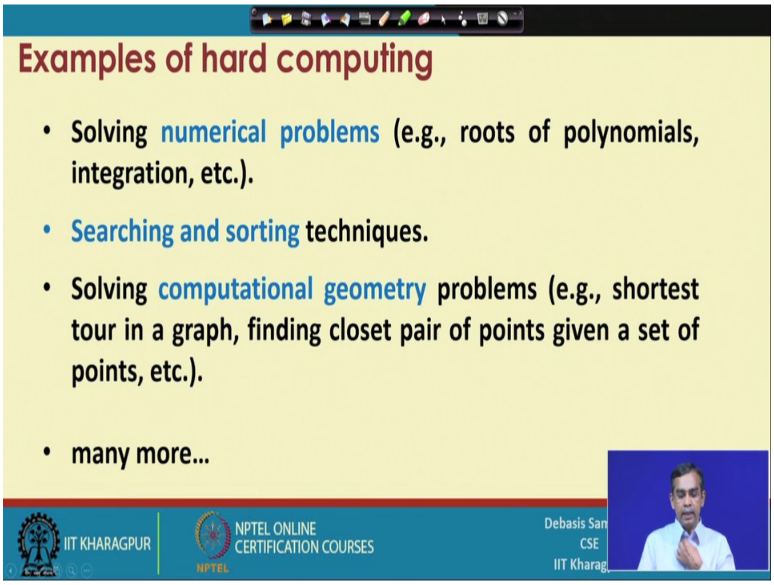
(Refer Slide Time: 03:21)



Now this is the concept of computing and this concept is first time coined by a mathematician is Lotfi Aliasker Zade. So, he is only termed as LAZ and he basically is the first person to introduce the concept of hard computing as a part of the concept of computing in general. So, according to LAZ the computing we can say it is hard computing if it provides precise result and the step that is required to solve the problem is unambiguous and then the control action; that means, those are the steps that is require is formally defined by means of some mathematical formula or some algorithm.

So, if a computing concept follows these are the three different characteristics then we say that computing is hard computing.

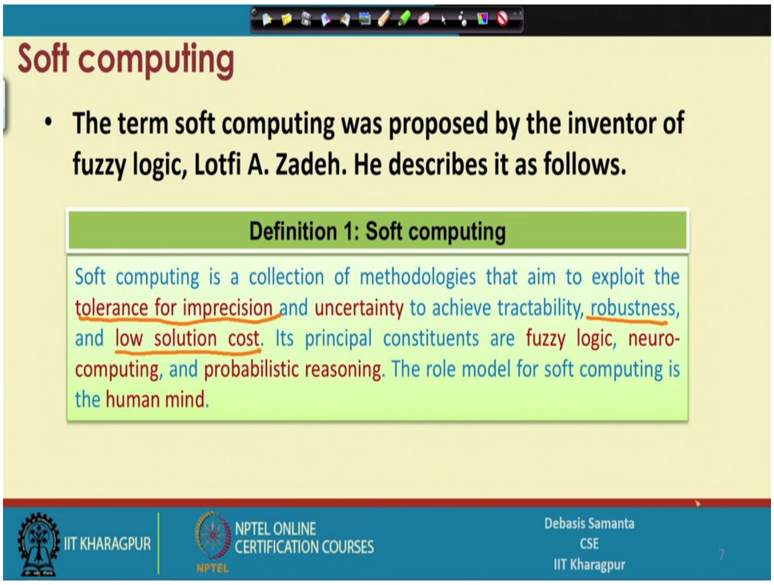
(Refer Slide Time: 04:27)



Now, I will come to some example of hard computing. We know in order to solve numerical problems for example, finding root of polynomials or finding an integration or derivation we usually follow some mathematical models and therefore, it is an example of hard computing. Now, searching and sorting techniques are frequently used in many softwares. So, these are the basically followed by some unambiguous steps and it always gives the precise result and it is basically defined correctly by means of an algorithm. So, it is an example of hard computing.

There are many problems related to the computational geometry for example, finding the shortest tour in a graph, finding closest pair of points given a set of points etcetera is basically is a task of hard computing. And there are many many such examples can be given. So, these are the concept of hard computing. And now, let us come to the concept of soft computing.

(Refer Slide Time: 05:39)

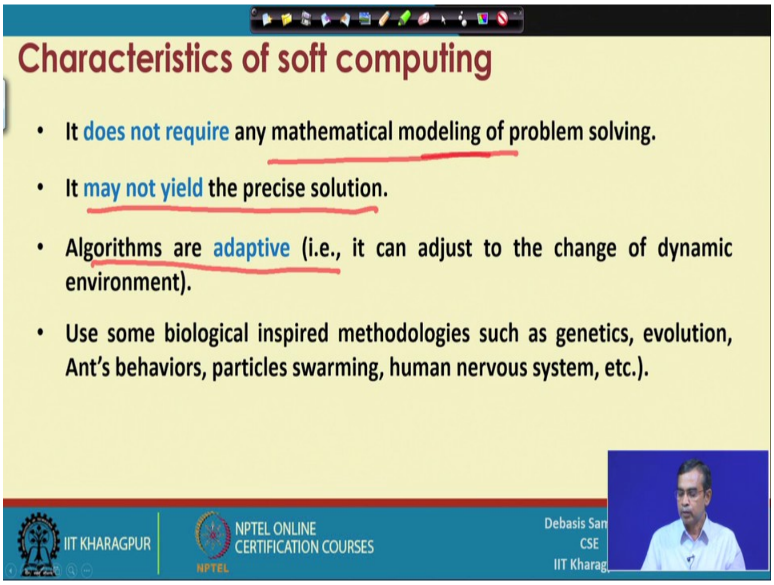


So, as I told you the hard computing first time proposed by LAZ. He himself also defined the concept of soft computing first times. According to him, the soft computing is defined as a collection of methodologies that aim to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness, and low solution cost.

Now, here I have underlined few things that you can mark it. So, the first thing it is basically tolerance for imprecision this is important. This means that the result that is obtained using the soft computing not necessarily to be precised and obviously, the result is uncertain. This because if you solve this problem several times it may give different result different time. And is a robustness, means it can tackle any sort of input noise including, so that is why it gives the robustness. And very important concept is called the low solution cost. Some problems if we follow hard computing then it is computationally expensive. However, if we follow soft computing then it is computationally very cheap; that means, we can find a solution in real time.

Now, so if this is the concept of some soft computing where the result is not necessarily to be precised, the step that needs to be followed is not necessarily the certain or unambiguous and then the result that can be obtained is also not necessarily to be same always. Then how this can be achieved? So, in principle the soft computing concept follow three computing paradigms. These are called fuzzy logic, neural computing and probabilistic reasoning. So, these are basically the soft computing paradigms and is basically these concepts the fuzzy logic, neural computing or probabilistic reasoning, if you see, this is the exactly the way the human can solve their own program. So, that is why the role model for soft computing is in fact human mind.

(Refer Slide Time: 08:23)

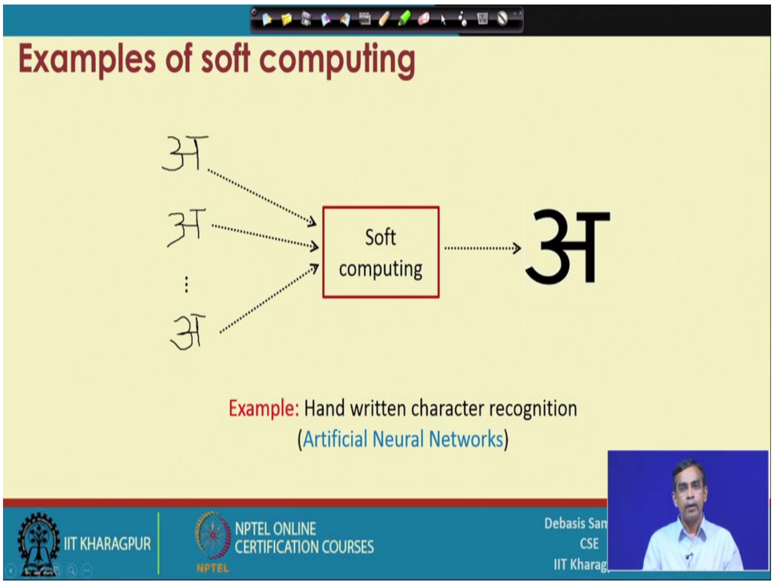


Now, let us see what are the different characteristics of soft computing? We have discussed little bit about it. So, soft computing is one concept of computing which does not require any mathematical model. So, it does not require any mathematical model or problem; that means, it does not necessary that an algorithm should be followed or the problem that we have to solve should be expressed in terms of mathematical formulation. And it may not yield the precise solution, the solution that is not that it will give you always the same or a unique. It can give time to time the different solution for the same problem even with the same input also.

But the solution is near about the accurate value. And algorithms are adaptive; that means, it can adjust to the change of any dynamical situation. I, by the means of dynamical situation, I want to mean that if the input is changes, suppose you want to solve one problem which require only two inputs, but later on the same problem require where twenty input is required. So, the same problem same computing concept can be easily adapted into whatever be the number of inputs are there, whatever the input values maybe there, whatever the other parameters that is involved in order to solve the problem.

Now, so I told you that a human mind is the role model behind the soft computing and actually it is some biological inspired methodology. So, that also constitute the concept of human behavior, such as genetics, the evolution, the behaviors of ant colony, swarming of particles, our nervous system, etcetera. So, basically the way the different natural phenomena works for us if we follow the same method and then try to solve our own problem this is basically the way exactly the use of soft computing to solve our own problem.

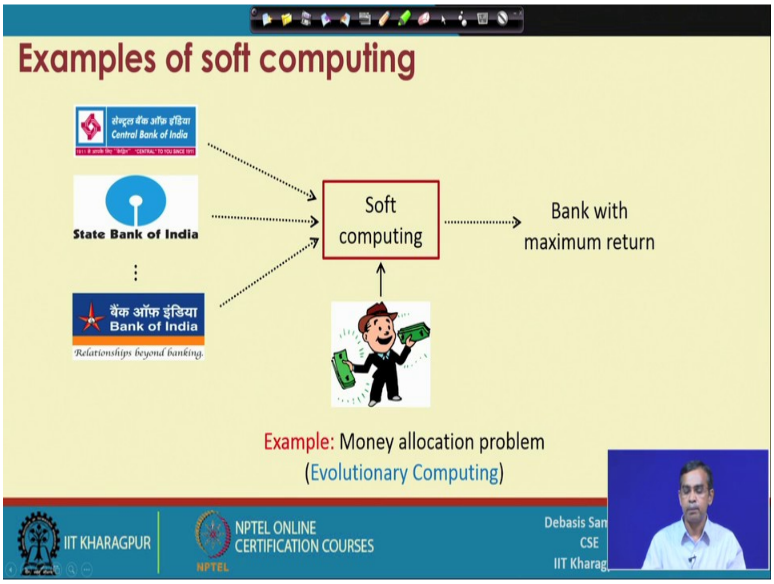
(Refer Slide Time: 10:35)



Now, I will give some example of soft computing so that we can understand how the soft computing can works for us. And in this example this is example is basically extracted from the hand written character recognition. Now, the different people if we collect the hand written character they can give the same characters in a different form.

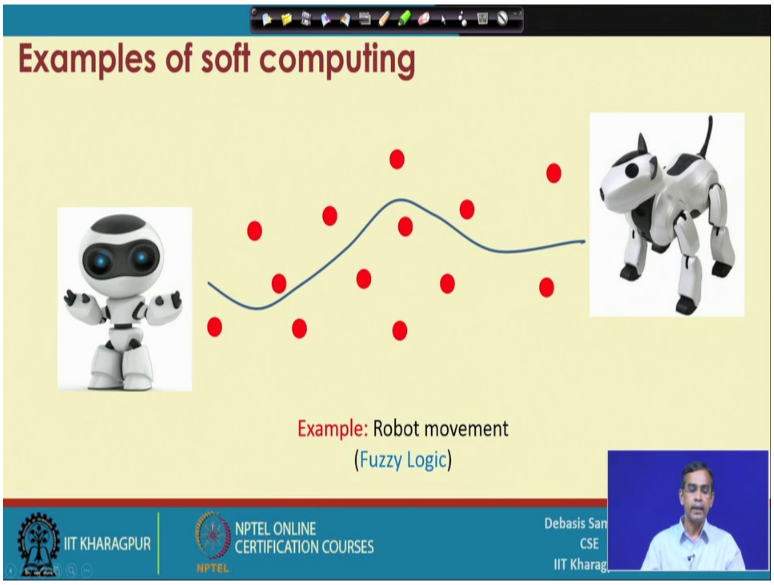
Now, even we know exactly whatever the different form or the way the people can write we can understand easily. For example there is a different way the input is given here and we can exactly step it here we exactly tell that this is [Aa]. Now, how it basically happens is basically we learn by the process that this is the letter resemble to a particular alphabets [Aa]. So, it is in the same way we learn it and this learned somehow stored in our memory and this is the learning phase and these learning basically works for us to recognize any unseen characters or unseen letters. And these basically the way actually our neural network our nervous system works and based on this concept the artificial neural network has been evolved and it is followed there.

(Refer Slide Time: 12:02)



Now, another example say, suppose a person wants to invest some money and for which the different the banks are available with the different policies, the different schemes and there is a flexibility for the person to invest all or some money into the different banks so that he can return the maximum profit. Now, here is the one problem that how we can store the, how we can invest the money in different bank so that we can get the maximum return. So, this concept is basically can be followed using some probabilistic reasoning or it is called the evolutionary computing for example, genetic algorithm can be followed to solve these kind of problem.

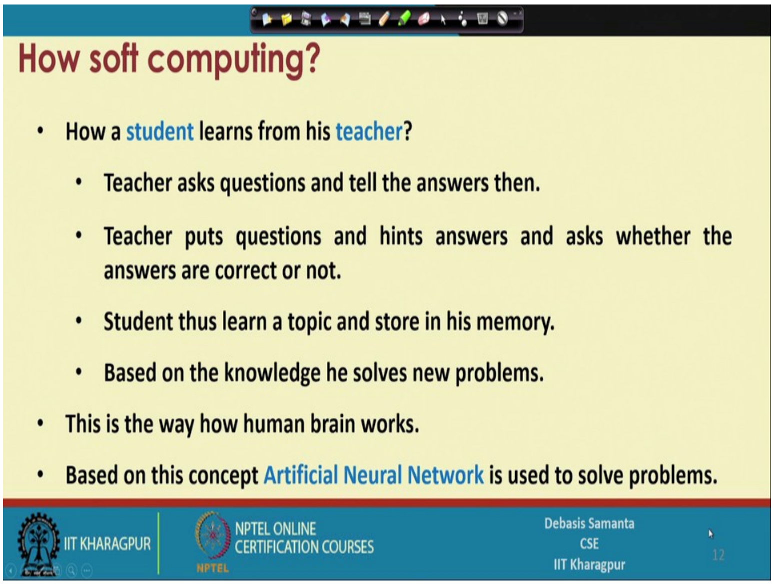
(Refer Slide Time: 12:59)



Another example it is from the robotics say, suppose one robot wants to move from here to these place and there are many obstacles are there. Now, so how the robots can calculate his movement so that without any collision, with any objects, he can move from his current location to target location within a shortest time. Now, this kind of problem, in fact, has lot of uncertainty or impreciseness, defining so for the input is concern. Because the robot is works like that. And then that kind of uncertainty can be solved using the concept called the fuzzy logic. So, fuzzy logic is an important parts of soft computing.

Now, here is again another question. So, we have discussed three different problems hand written character recognition, allocation of money into the different banks and then movement of the robots in three different corners. Now, the first example that we have discussed that it is the problem which can be solved very effectively efficiently using artificial neural network. The second problem that we have discussed it basically solved using some probabilistic reasoning and it is basically one problem called evolutionary computing or genetic algorithm. The third problem that we have discussed it is basically the fuzzy logic, how the fuzzy logic can be exercised to solve some problem where lot of uncertainty involved.

(Refer Slide Time: 14:48)



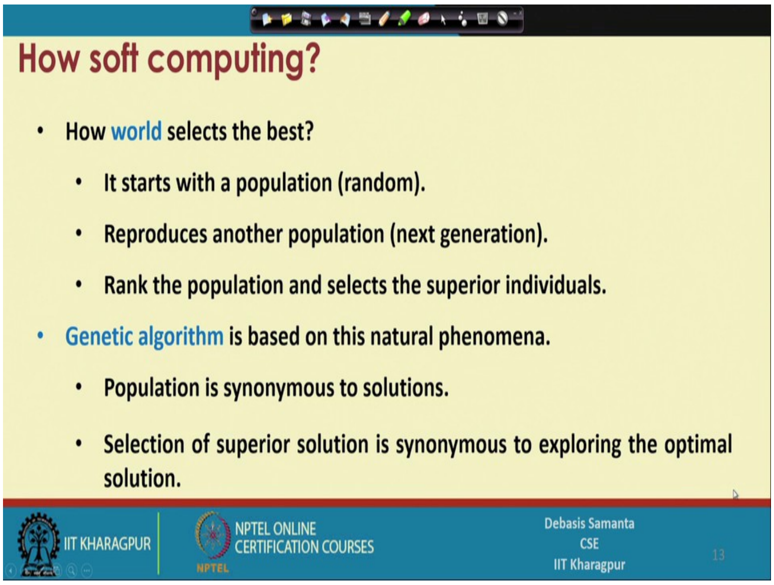
Now, I want to, now discuss about the different techniques that can be followed or the different techniques which is basically behind the above concepts. For example, how a student learn from his teacher? Here the two part is involved one is the student and is the teacher. Now, consider student is basically a computing machine and teacher is basically once two I mean gets some output for a given input like. So, how a student is learn from his teacher or basically how such a system can be developed, here the teacher is responsible to develop a system and here system is student.

So, usually teacher ask questions and tell the answer. Then there is another way teacher puts some questions and hints an answers and ask whether the answers are correct or not, students here basically to check whether the answer correct or not. Students then, students thus learn a topic and store in his memory. So, basically by the process if we discuss several time the same different questions, different answers, different questions, hints to the different answer for the same question or different answers for the different questions.

So, students listen to those and then by the process learn a topic and whatever the students learns it basically store in his memory. Now, based on the knowledge he then can solve many new problems assigned to him. So, basically it is a concept of learning how to learn something and then based on this learning how he can solve the problem.

So, this is the way exactly our human brain works in fact. And based on this concept the artificial neural network is used for example, hand written character can be recognized.

(Refer Slide Time: 16:45)

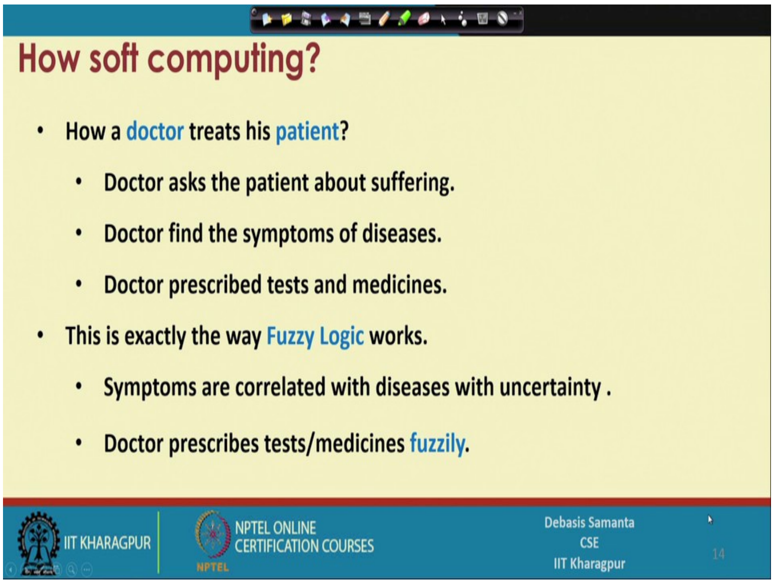


Now, another example, so how world selects the best? It is basically a natural process. So, in this process is basically starts with a population and initially it consider a random population. So, when our worlds evolve first time it is started with some population and is a random population. Random means whatever the objects these are possible there and it then reproduces, reproduces to develop another population, we called it is a next generation. And then all the population that we obtained so we rank them and select the superior individuals. So, here basically population generation, then reproduction and reproduction followed by the ranking and then it basically selects based on this ranking the best individuals. So, basically best population or best solution.

Now, the concept of genetic algorithm is based exactly on the same phenomena, it is called is basically genetics. And here in this context the population is synonymous to solution. So, we can start with some random solution those are not necessary to be an optimal and then we have to reproduce from this set of solution another solution and then select the best solution. The same thing can be repeated several times ultimately until we can achieve the best result. Now, here selection of superior solution is synonymous to exploring the optimal solution.

Now, here we can see all the method that can be followed in a probabilistic manner or in a randomized sense. So, that is why the genetic algorithm follows a probabilistic reasoning to solve problem particularly solving optimization problem.

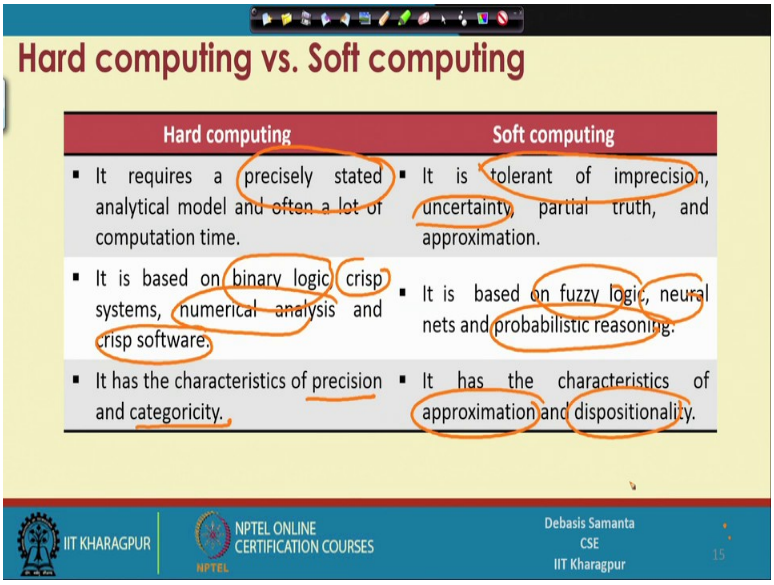
(Refer Slide Time: 18:45)



Now, as another example how a doctor treats his patient? Here doctor is a one party and patient is another party. Now, patient wants to solve his problem with the help of doctor. So, doctor is the computing system in this case. So, usually it works like this. Doctor asks the patient about the problem that he is suffering and doctor find the symptoms of disease from the patients input, and then doctor prescribe some tests and medicine. This is the exactly the way the fuzzy logic works.

So, fuzzy logic take some input which is basically related to solving some problem and then based on this inputs he predict certain output. So, here symptoms are correlated with disease and you know whatever the disease doctor will guess or patient will tell they are basically not a certain; there are some uncertainty with the input. So, this is why it is called the symptoms are correlated with disease uncertainty and then the doctor prescribe medicines or whatever the test it is also fuzzily; so that means, with certain uncertainty. So, fuzzy means it is uncertain in this sense.

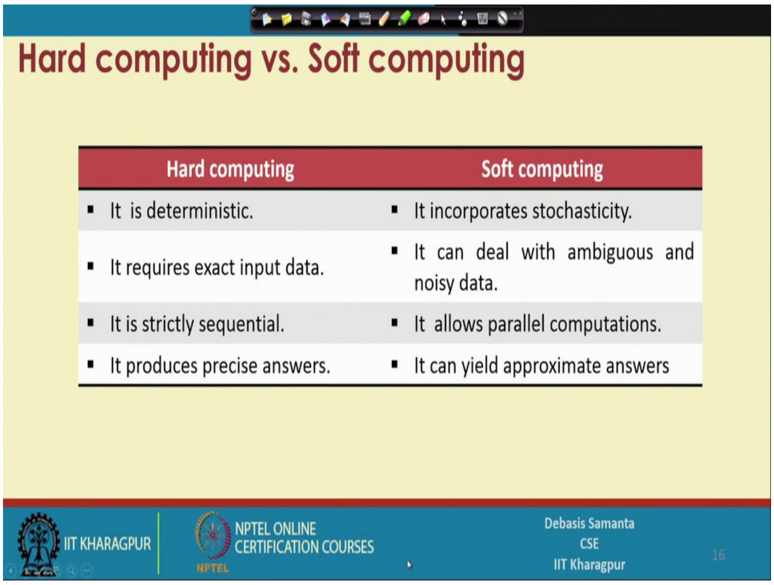
(Refer Slide Time: 20:11)



Now, let us discuss about hard computing versus soft computing. Now, so for the hard computing is concerned it requires a precisely stated analytical model and obviously, it is a computational expensive on methodology. On the other hand soft computing it is imprecision, tolerance to imprecision means we can be happy with some solution which is not exactly the precise one and with uncertainty, the partial truth and approximation may works for us. Only the requirement that is that the problem which cannot be solved using hard computing in real time can be solved using soft computing in a real time.

The concept of hard computing basically based on few concept called the binary logic, crisp system, numerical analysis and some crisp software; the software basically if run for the same input it always give the same output. Whereas, the concept that is followed in soft computing is based on the fuzzy logic, the neural networks, and probabilistic reasoning which is totally different than the concept that is followed in hard computing. And so, hard computing basically has the characteristics of precision and categoricity, it works for a certain kind of input and it works well for that input. Whereas, the soft computing is a characteristic of approximation; exact result is not required, but it can be near accurate result and dispositionality; that means, it can be applied to varieties of input, the different type of input, different number of inputs as well as.

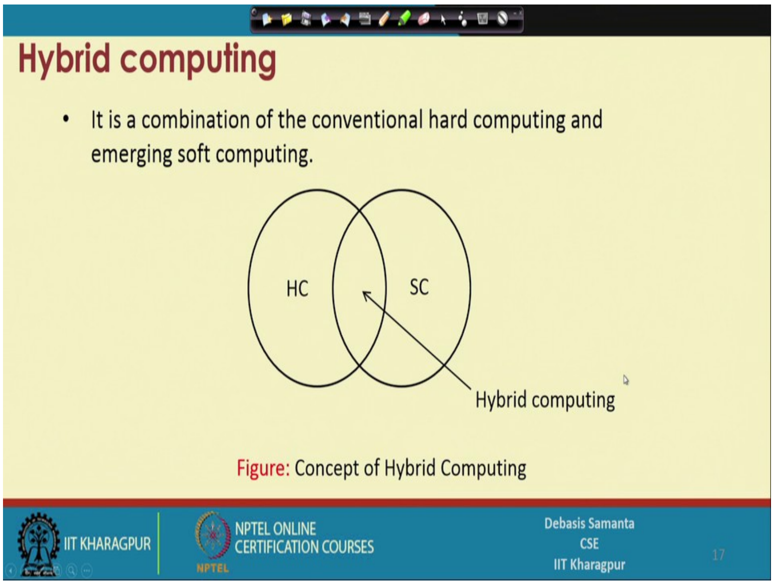
(Refer Slide Time: 22:09)



Now, another differences another differences between hard computing and soft computing it is deterministic whereas, the soft computing is stochastic means probabilistic. It requires exact input data in case of soft computing it basically requires an ambiguous and noisy data.

Hard computing usually is followed strictly sequential methods; however, the soft computing can be carried out using parallel computation. Hard computing produces precise answers whereas, soft computing can yields approximate answers. So, these are the differences between hard computing and soft computing and I hope you have understood the difference between the two.

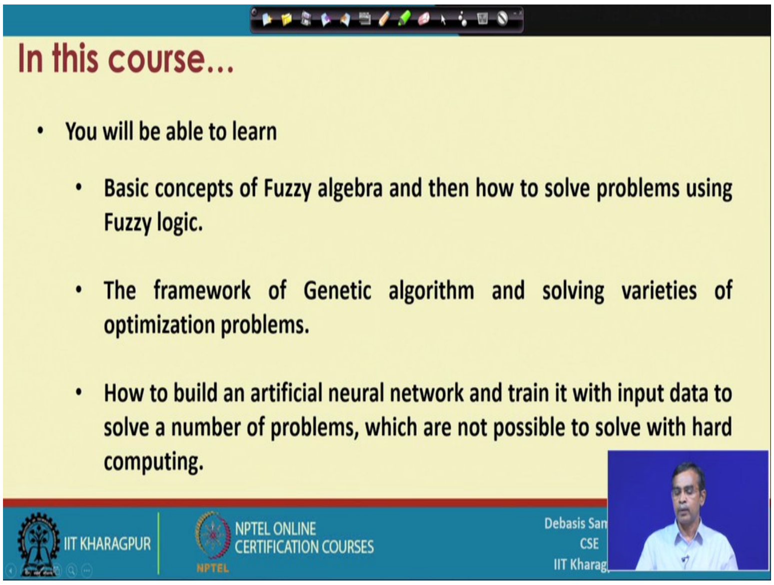
(Refer Slide Time: 23:05)



Now, so there is hybrid computing. It is basically combination of the two into solving a particular problem. So, few portion of the problem can be solved using hard computing for which we have a mathematical formulation for that particular part and then where we required a precise input. And there are some part of the same problem maybe which cannot be solved in real time for which no good algorithm is available and we also do not required accurate result some near accurate result is sufficient for us then we can solve soft computing for that part and then mixing together is basically the hybrid computing.

So, if we know hybrid hard computing, if we know soft computing and if we know some problems where whatever the characteristic involved to solve either hard computing way of soft computing way we can inter mix the two approaches and then hybrid computing can be obtained.

(Refer Slide Time: 24:14)



Now, in this course you will be able to learn basic concepts of fuzzy algebra and then how to solve problems using fuzzy logic. Then we will be able to learn the framework of genetic algorithm and solving varieties of optimization problems. And then how to build an artificial neural network and train it with input data to solve a number of problems which are not possible solve with hard computing.

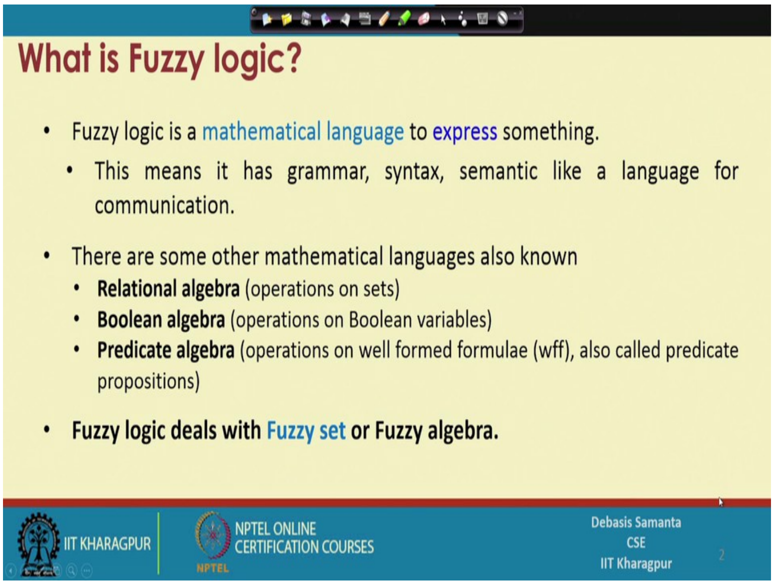
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 02
Introduction to Fuzzy Logic

We will start our lecture. This is the beginning of the topics, the fuzzy logic. So, fuzzy logic is an essential component for the soft computing. So, today we will learn about basic concept of fuzzy logic and to understand the fuzzy system we should familiar our self with different terminologies. So, we will explain the different terminology related to the fuzzy logic.

(Refer Slide Time: 00:47)

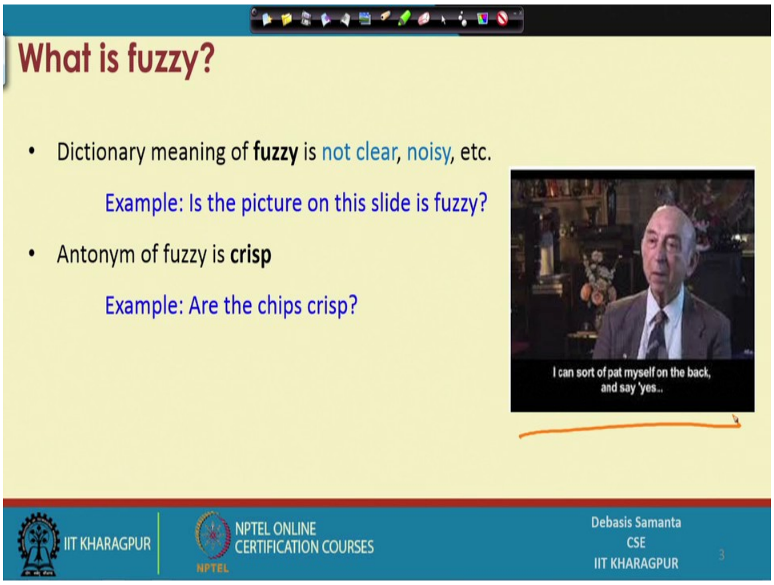


So, the first is what is fuzzy logic? In fact, fuzzy logic is a language we can say more precisely is a mathematical language like any language you know. So, this language is also used to express something which is meaningful to others. So, it is a language this means that, it has grammar, it has it own syntax, the meaning like a language for communication like English.

Now, like fuzzy logic there are many mathematical languages we know. So, one language is called relational algebra which is based on the operations on set. So, this is also called relational logic. Boolean logic is basically based on the operations on Boolean variables it is Boolean algebra it is also called. And predicate logic or it is called the predicate algebra, which is basically operations based on the well-formed formulae or proposition also called the predicate propositions. It is very interesting that fuzzy logic like relation logic, Boolean logic and predicate logic, it also deals with some elements; the elements on which this fuzzy logic depends is called fuzzy set and it is also alternatively call the fuzzy algebra. And another interesting fact is that the fuzzy logic essentially combined the different algebras like relational algebra, Boolean algebra and predicate algebra together.

So, it is basically a mixture of the different mathematical languages to define another new language that is the fuzzy logic.

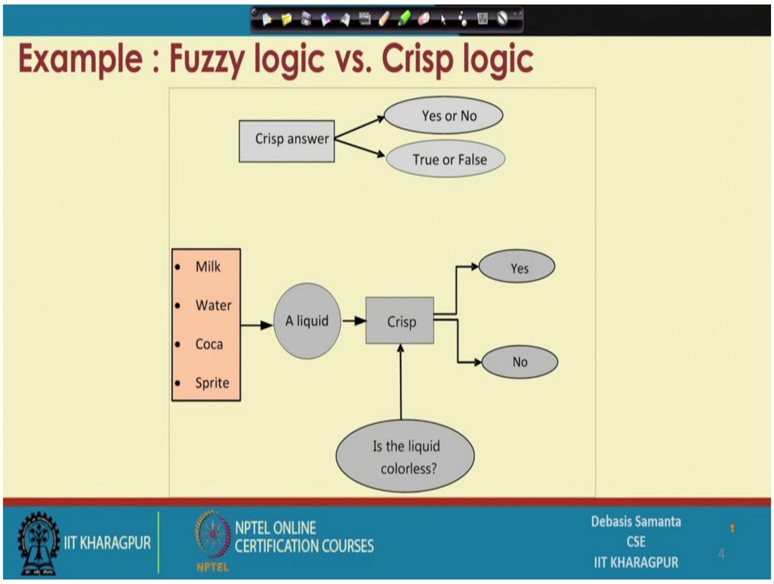
(Refer Slide Time: 02:47)



Now, so the word fuzzy may not be new to us. So, if we search dictionary the meaning of fuzzy is not clear or it is called noisy it is like that. Now as an example so we see one figure here we see one figure here this is the figure now. So, sometimes we see that is the picture on this slide is clear. So, we can say yeah the picture on this slide is fuzzy; that means not clear whatever the wording it is clear we may say that it is not clear or there are many noise in the image. So, the image is noisy, image is fuzzy.

In other words we can understand the meaning of the fuzzy if we see it's antonym. The antonym of fuzzy is crisp. Now crisp in the sense that if we say there is a there are 2 regions and if we say the boundary if the boundary is not clear then we can say the 2 regions are separated fuzzily, on the other hand if there is a strong boundary by which we can easily distinguish 2 regions clearly then we can say that the boundary is crisp. So, this way we can understand the fuzzy versus crisp. We learn many thing about this fuzzy versus crisp in our next slides next discussion.

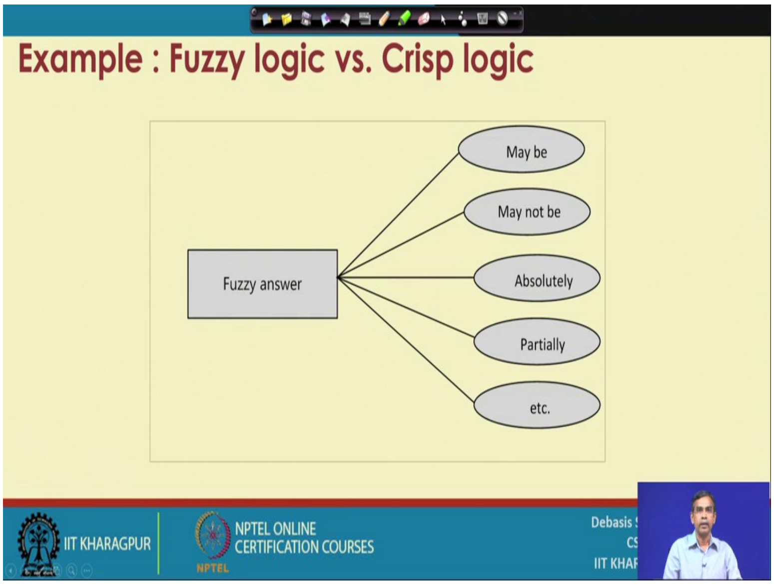
(Refer Slide Time: 04:41)



So, we have a little bit understanding about the meaning of fuzzy. Our next discussion exactly with some examples. So, how the 2 logic may be say logic with fuzzy sets and logic with crisp set. Ok. So, I say here set. Anyway, so I will discuss about what exactly the crisp set it is anyway. So, fuzzy logic versus crisp logic. Now say, if we ask some questions and then answer to that question if has the clear meaning then we can say that answer is having crisp answer.

So, crisp answer usually expressed in the form of either yes or no, true or false, like this. As an example, suppose the question is that we have to identify a liquid now, any liquid like milk, water, coca, sprite is given and then if we ask the question that is the liquid colorless? Now you have to give the answer in terms of only 2 things, yes or no. Then it is called the crisp answer. So, this way we can understand exactly the crisp crisp system.

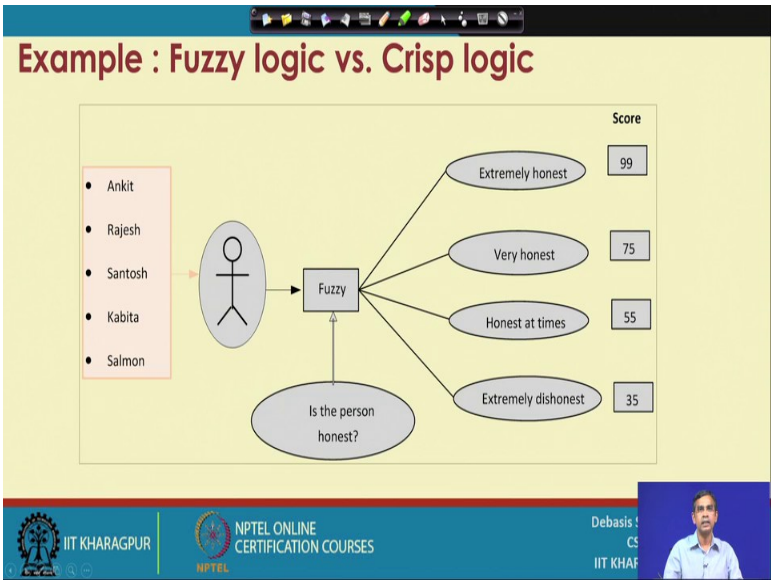
(Refer Slide Time: 05:59)



And alternative to crisp system the fuzzy answer let us see how the fuzzy answers can be like.

So, if we ask one questions and the answer can be of many instead only 2 solid answers. So, the answer may be may be, may not be, absolutely, partially, etcetera. So, there are many many form, many values for the same answer.

(Refer Slide Time: 06:29)



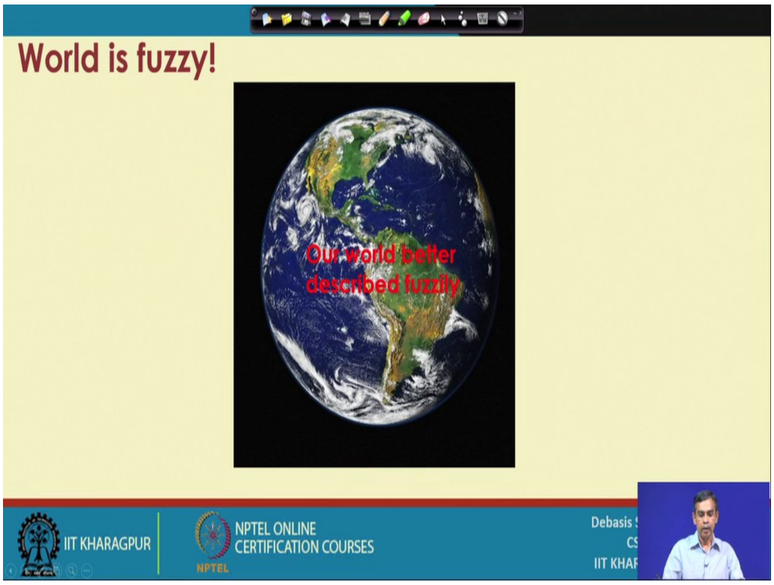
So, this is basically the concept of fuzzy answer. I can illustrate the concept with an example. Say, fuzzy system is like the question is, is the person honest? And a person is given as an input let them are the Ankit or Rajesh, Santhosh, Kabita, Salmon like.

So, if we ask the question is the person honest say Ankit. So, their answer may be extremely honest, very honest, honest time to time or extremely dishonest. Now so for a for a question unlike the crisp question, for the fuzzy questions, or for a question the fuzzy answer, on like the crisp answer may have different what is called the answers. Now so, this different answers if it is there for the same question then which is the correct answer actually. Because all answers seems to be acceptable or rejectable. Now which answer is there?

Now, we can give a score to each answer. So, here I have given a score here for each answer. For example, extremely honest 99, very honest 75, honest at times 55, extremely dishonest 35. So, this means that if it is a 2 valued answer like say crisp answer then only 2 and then score will be on 100 and another is 0 whatever it is there. But here the different values of the score. This means that the answer which is a very honest it is also the correct, but correct with a validity score it is called the 75.

Now obviously, question that arise that, how we know what is the score actually? So, we will discuss about that how the score for a for an answer can be calculated and that can be tagged into that answer to signifying that how answer is significant or how answer is acceptable, so far the question is concerned. Anyway so the idea is that, these are the answer is called the fuzzy answer for a given question unlike the crisp answer.

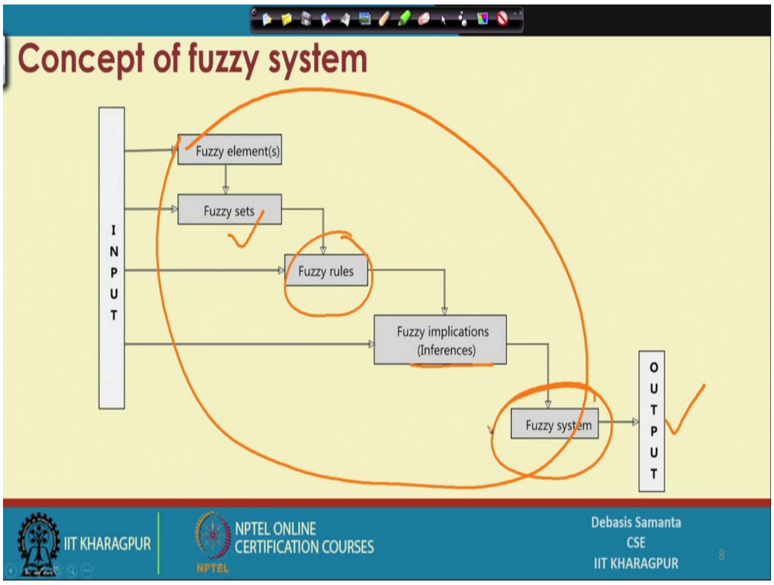
(Refer Slide Time: 08:41)



Now, in fact our world can be better described fuzzily. This is because if I say what is the temperature today? So, you can get an answer like very hot, some people can say that comfortable, the extreme or very cold it is like this. So, this means that for the same question, the answer can be different if the same question is fired to many people. But everybody can give the answer according to their own estimation whatever it is there, but the answer is like that. Like the temperature another what will be the weather today? So, if I ask to predict it to some expert person then he will give the answer fuzzily; that means, yeah weather is sunny today, may be sunny, may not be sunny, may be cloudy or it is like that.

So, the answer can be for same questions of different form and different form has their own value and then we have to take all the values, in fact, and then process it. So, that the answer is acceptable to us.

(Refer Slide Time: 09:52)



Now, so the best idea it is that the system that we are using it is basically better can be described fuzzily or it is basically the system or we can say that everything is in the form of a fuzzy and then we can take the fuzzy manners or the fuzzy way to describe any system rather.

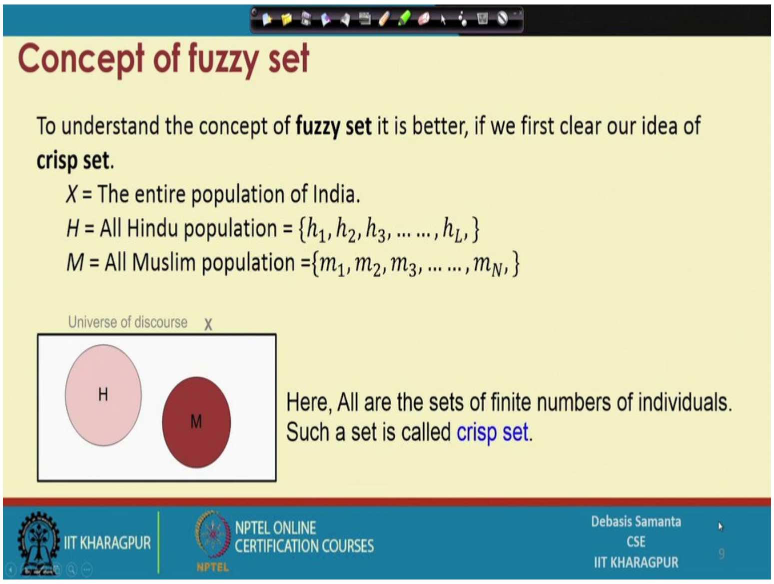
So, if we describe a system in the in the way of in the way the fuzzy decides then this system is called the fuzzy system. Now typically the fuzzy system has many ingredients or elements. So obviously, the input and output is the part of any system that we have already discuss in the first lecture itself. So, if this is the entire system then these are input and then output, now need not to say that input is obviously in the form of a crisp. Because we usually give input to the system in the form of a crisp value.

Similarly, the output also should be in the form of a crisp value. So, in this system there are 2 boundaries one input and output. Input and output are in the form of a crisp value. However, input can be transformed into some fuzzy form and then the fuzzy system can come into play. And so in this type of fuzzy system there are many constituents many elements are there.

So, the first element is called the fuzzy elements and taking one or more fuzzy elements we can discuss about the fuzzy set and then many fuzzy sets can be connected with the set of another element it's called the fuzzy rules and finally, a set of fuzzy rules can govern us to decide is called the fuzzy implication or it is called the inferences and these whole the things constitute what is called our fuzzy system.

In other words, to understand the fuzzy system, it is our task to understand what exactly a fuzzy element it is. And then what is a fuzzy set and then using the fuzzy set how the fuzzy rules can be obtained. And then how the inferences can be described in the form of a fuzzy rules. And that all these things if we learn it, then we will be in a position to discuss about the fuzzy system. So, in our subsequent lectures we will basically discuss about all these elements one by one. Today will discuss about the fuzzy elements first.

(Refer Slide Time: 12:33)



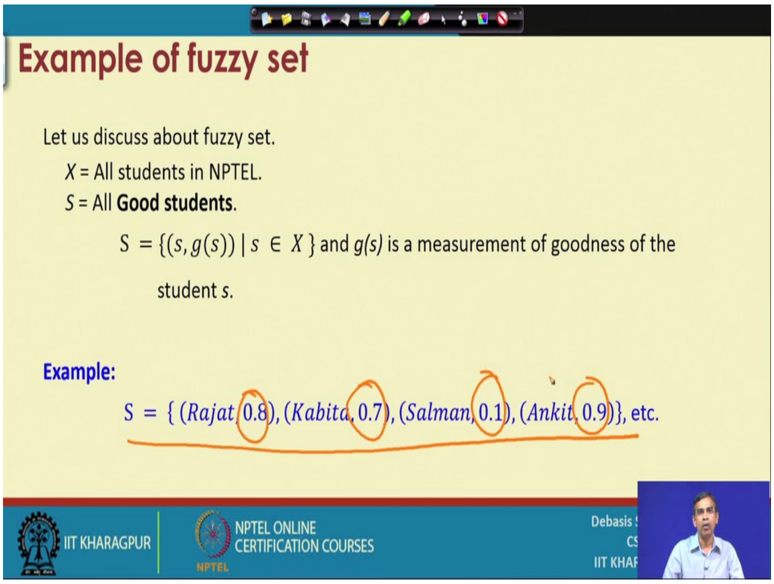
Now, let us see exactly what is the fuzzy elements. So, fuzzy elements basically essentially it a fuzzy set. So, we can better describe a fuzzy set in the form of a crisp set actually. So, we know exactly the concept of set. So, this the traditional set that we know it is, in fact, is a crisp set. For an example, say X denotes a crisp set and it denotes the entire population of India. Then you can say what are the elements? Yourself, myself are the elements belongs to the set X.

Now, I can derive one set again from this set X or some other means suppose it is the H. H is the another set it denotes all Hindu population. So, any element that is means any person or any individuals belongs to this set is the set composition itself. For example, here h_1, h_2, h_3 all these things are the elements to this set they are basically individual who basically satisfy some characteristics being Hindu population. Like Hindu population we can define another set say all Muslim population for example, these are the set of all Muslim individuals.

So, these are the example of crisp set and we know any crisp set can be better describe in the form of a graphs or it is a venn diagram. So, we have we have shown one venn diagram here. For this H, M and X whole the things are basically shown here and we can see that there are the 2 boundaries. The 2 boundary essentially difference or basically define solidly the 2 regions. One region belongs to H and another region belongs to M and these 2 regions a basically belongs to another bigger region.

So, this bigger region is basically called universe of discourse in this case it is X. So, all the regions whatever it is there has a solid boundary and that is why they called the crisp set.

(Refer Slide Time: 14:47)



Now, so like crisp set the fuzzy set is also almost similar, but little bit difference is there; difference so for the presentation is concerned. For an example, so suppose X, X denotes a set and let the set be all students in NPTEL. So, this is the universal of discourse in this case.

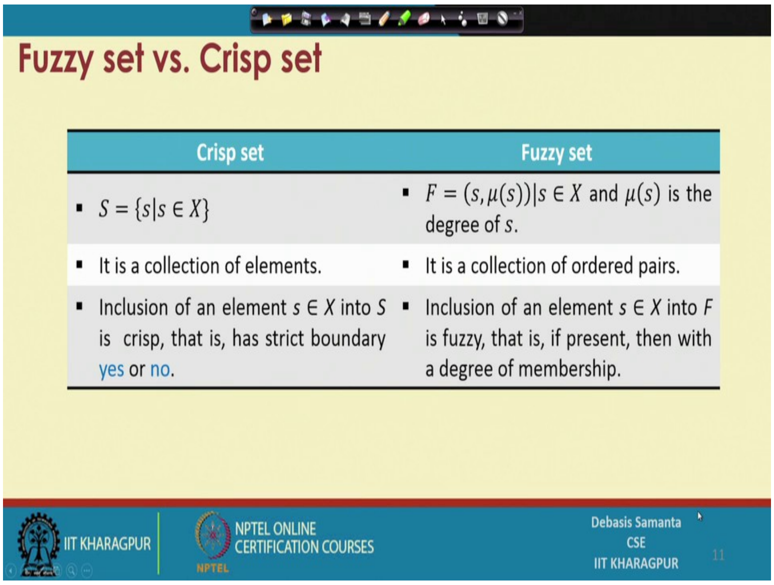
Now, I let us define one set belong to this X, let this set be S. And we define this set S as all good students. Now let us see how the same thing can be defined using a fuzzy manner. So, we define the set S as a 2 things in each elements; one is s the elements itself and g(s) some measurement of S itself, where s is any element belong to X and g(s) is a measurement. Now this measurement, in fact, we can say measuring the goodness of a student.

Now, for example, if I want to measure the or evaluate a student. So, how I can evaluate? I can take some exams and I can take the marks obtained by the students in that exam. So, g(s) can be same type of measurement like. So, it is a goodness measure. It is or rather it is called the measurement that that the s belongs to the set S. So, for example, here again we can see. So, suppose there are few students which are Rajat, Kabita, Salman, Ankit like and their measurement is expressed here for a Rajat is having the score 0.8, Kabita having 0.7, Salman 0.1 and Ankit a 0.9.

So, this set signifies that all students who belongs to this set like Rajat, Kabita, Salman they are the good student, but goodness is defined by means of measure. In other word Salman if he is a good student then Ankit is also good student. But Salman being a good student his score is 0.1 and Ankit his score is 0.9. So, the difference between the two is basically how they have their own membership values; that mean, 0.1 0.9 whatever it is there, but all them belongs to the good student in fact. All though Salman may scoreless or Ankit may score highest here. All of them are the good students belongs to the good students actually.

Now, here another point you can note that the measurement value that we have mentioned here is basically in between 0 to 1. Actually it is the concept that is followed in fuzzy logic all the measurement value g(s) like. So, value should have in 0 to 1 both inclusive. So, any value in between 0 and 1 are the basically taken as the membership value for this one.

(Refer Slide Time: 18:06)



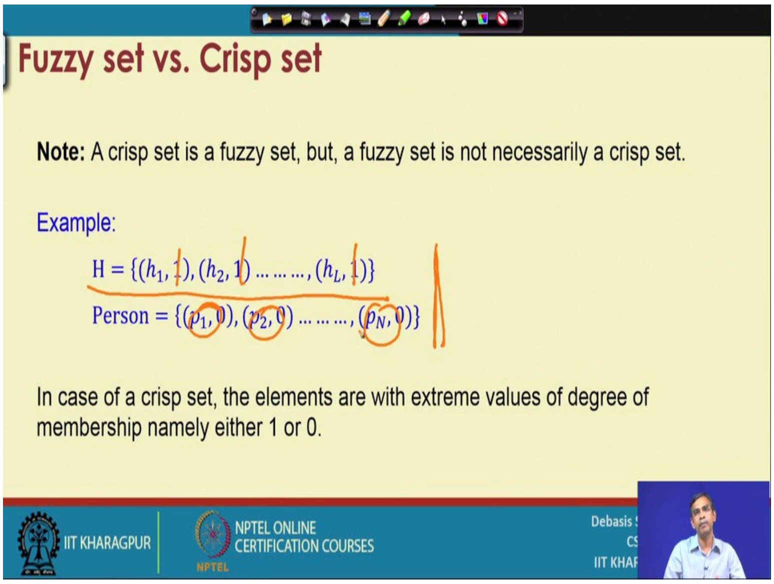
Now, so we have a little bit understanding about the fuzzy sets and let us see what are the difference, the salient differences between the crisp set and the fuzzy set. So, the differences between the two sets are defined in the form of a table. So, if we define a crisp set it is basically is a collection of elements; that means one part only, so s. Where as if it is a fuzzy set then it is a collection of ordered pair it is called. The first part is the element itself and second part is the measurement of that element itself.

So, sometimes this measurement μ(s) in fuzzy theory it is called the degree of S or it is also called membership value of S. So, what you have understood is that a crisp set is a collection of elements whereas a fuzzy set is a collection of ordered pairs. So two things together form one element in the fuzzy set.

Now, inclusion of an element, any element say, s into the set a S capital S is crisps, that is it has strict boundary, yes or no. If that element belongs to the set yes or no we can easily justify that one. However, inclusion of an element s into F the which is a fuzzy set is present then with a degree of membership. In other words a same element say, s can belongs to two fuzzy set F and G, but with different membership values. For example, if F denotes the good student and G denotes the bad students then same element say, s can belongs to the good student as well as bad student, but with different membership value. For example, s appears in F with membership value 0.7 whereas; the same element belongs to the set G with membership value say 0.3. So, it is like these.

So, same elements may appear into the two sets with different membership values whereas, same element may not appear into two crisp set, it is either in one set or another. So, there may be ok.

(Refer Slide Time: 20:27)



So, we have understood about few definition about the fuzzy set versus crisp set. Now we will discuss about, one point you can note is as, I already told you the membership values or degree of membership value we can say alternatively like this. Degree of values, degree of membership values or membership values that can be their if an element belongs to fuzzy set with any value 0 to 1, inclusive and then any value in between 0 to 1 inclusive.

On the other hand, the same element actually if it is a crisp set can also express in the form of a fuzzy form with the membership value 1 and 0 only. For example, here is the set H. So, if that element presents there, then I can say the degree of membership one. If the element does not belongs to that set then we can say the degree of membership is 0. So, basically this is a fuzzy set essentially with the membership value 0 and 1. Now with this understanding if we do not write this one this one and this one then we can say that h is a crisp set which elements are h_1, h_2, h_L.

On the other hand if it is the membership value 0; that means, this element does not belong to this set. So, in this case the Person becomes a null set. So, basically 0 and 1 being the two extreme values can be expressed to define a crisp set in the form of a fuzzy set. So, this way we can say that a crisp set is a fuzzy set, because anyway crisp set can be converted in the fuzzy set easily. But a fuzzy set cannot be expressed always in the form of a crisp set. Because their membership value not necessarily always 0 and one in between 0 and 1.

So, this is the one conclusion that we can infer it from our discussion that the crisp set is a fuzzy set, but a fuzzy set is not necessarily a crisp set.

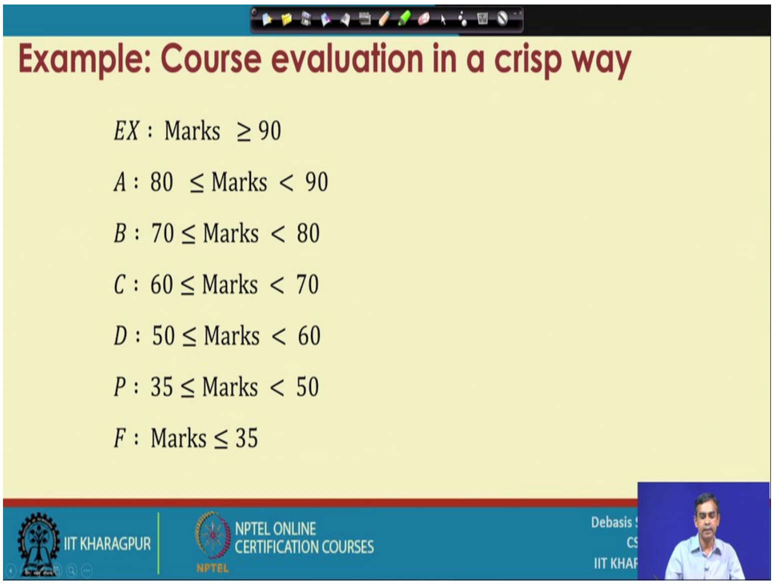
(Refer Slide Time: 22:31)

So, we have understanding about the fuzzy set versus crisp set or we can say crisp logic versus fuzzy logic little bit. Now let see one important point here, so far the fuzzy set decision is concerned the membership value and there is a question that how the membership value each elements can be decided and who can decide this membership values for each elements which belongs to the fuzzy set. I can give an example.

Say suppose all cities in India or more precisely say suppose there 6 cities in India like Bangalore, Bombay, Hyderabad, Kharagpur, Madras, Delhi right and I want to define one set let the name of the set is city of comfort. Now I have decided some value let the values for each set belongs to the set like Bangalore is 0.95 and so on. Now so the idea is that how this comfort of the Bangalore city 0.95 can be decided. There are certain population vote or something like population opinion or any way feedback whatever you can consider by this feedback, if we normalize those feed back into the value this one then it will give us to a fuzzy values.

So, this way we can have the fuzzy membership and regarding the membership value we will discuss in details in due time.

(Refer Slide Time: 23:58)

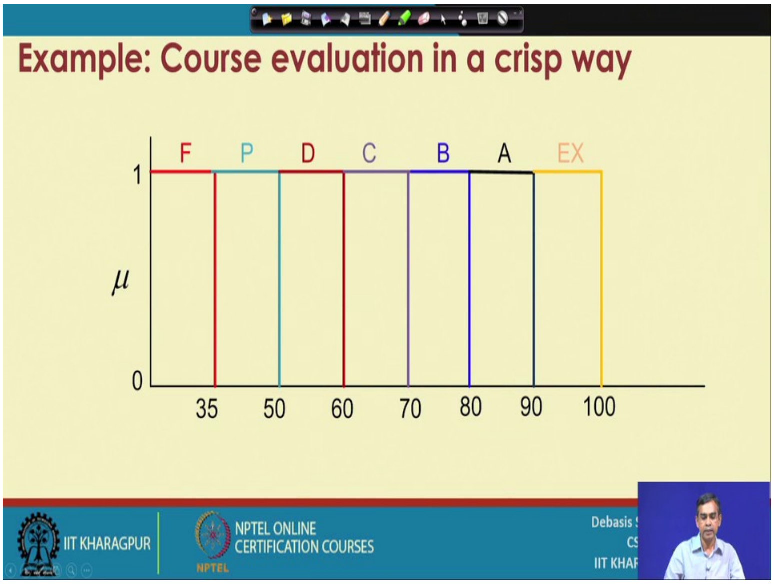


Now there is another example which I would like to mention here. So, that we can understand the concept of crisps versus fuzzy. The idea it is here, say we know exactly how to grade the marks obtain by a students in a subjects.

So, basically this is the grading formula, now these are the grading, that we can we can see that there is a strict boundary between one marks to another. So, a marks will be either belongs to the grade A or it is EX or B, but cannot be a same marks belongs to the 2 different grade. For example, one mark which is there in this it can belong to this one, that mean the marks can be EX, marks can be A, marks can be B, if it is marks in EX it is definitely with certain membership value is a 0.2 if it is belong to B then maybe it is 0.3 if it is belongs to A then maybe it is 0.9.

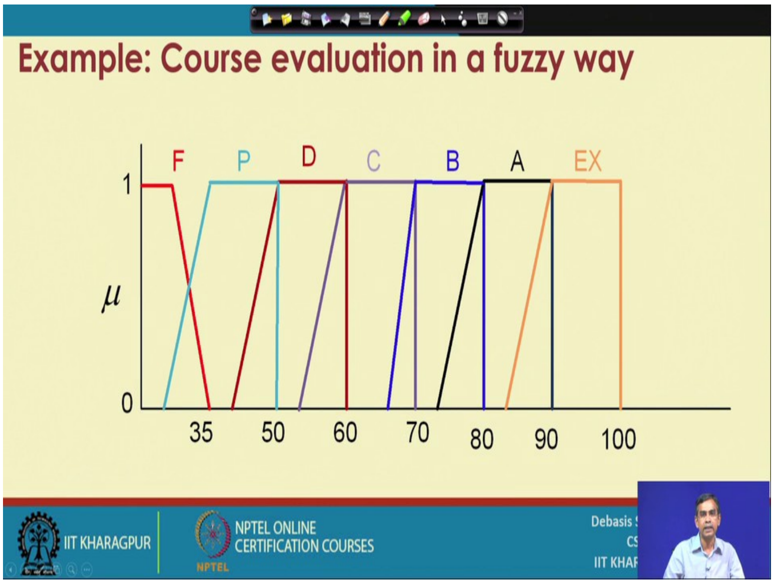
So, there is a there is a concept and this is basically the example of crisps formulation for the marks. Now the same thing if we do it in a fuzzy formulation it will look like this.

(Refer Slide Time: 25:14)



So, this is basically the graphical display of the crisp formulation and the fuzzy formulation we can see it is the fuzzy formulation.

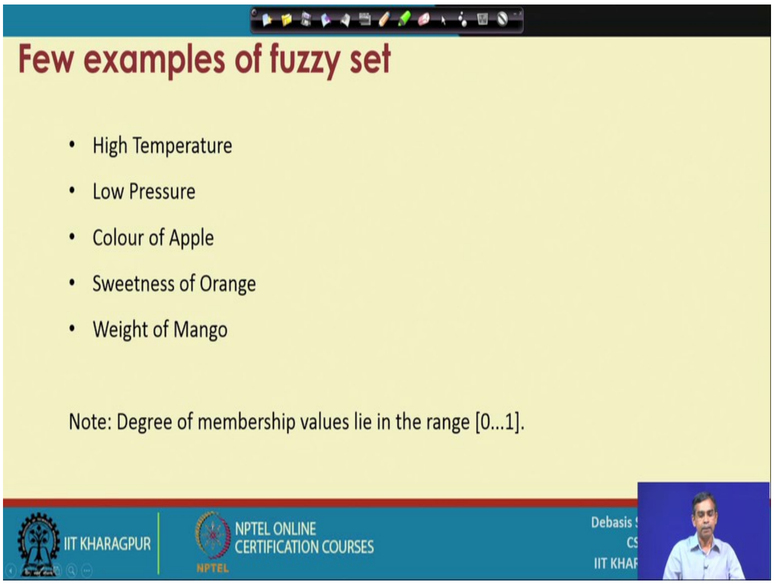
(Refer Slide Time: 25:22)



So, here we can note that any marks for example, any marks say it is this one this is the marks. So, this marks is basically these basically denote the D grade and these basically denotes the P grade. So, this marks both belongs to the P grade and both belongs to the D grade. If it is if we draw like this so if it is a D grade then this is the membership value and if it the P grade then this is the membership value.

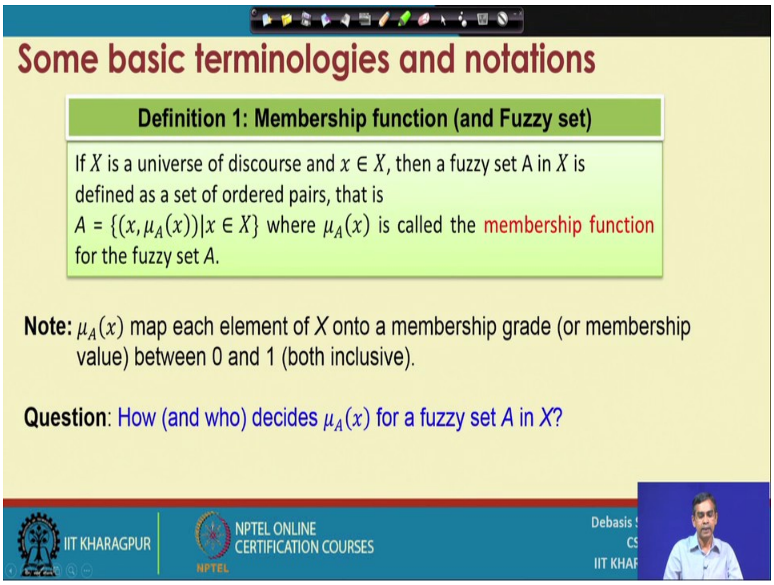
So, the same marks belong to the two sets P or D with the different membership values.

(Refer Slide Time: 26:03)



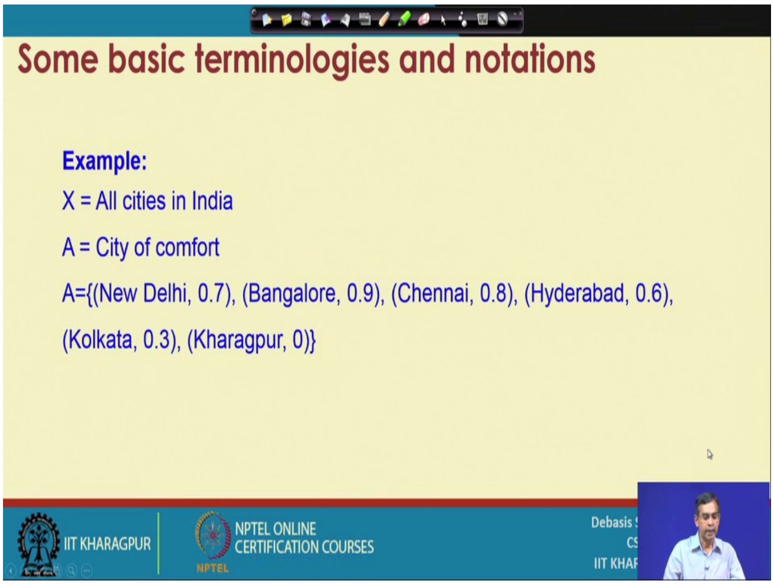
Some examples further can be used for example, temperature is high. So, we can discuss it with is in a fuzzy form low pressure, color of apple, sweetness of orange, weight of mango and so on so on. So, these are the few examples which basically we know. So, these are the input and then they can be discuss in a fuzzy form also.

(Refer Slide Time: 26:29)



Now, for the definition we will start with few terminologies. So, the first that a membership function and so it can be defined like this. If X is a universe of discourse and if any element x, which is belongs to this X, then a fuzzy set A which is defined in X is defined as a set of ordered pairs, as I told you ordered pairs x and μ(x). So, this is the concept of fuzzy sets and definition of basically membership function and these fuzzy set.

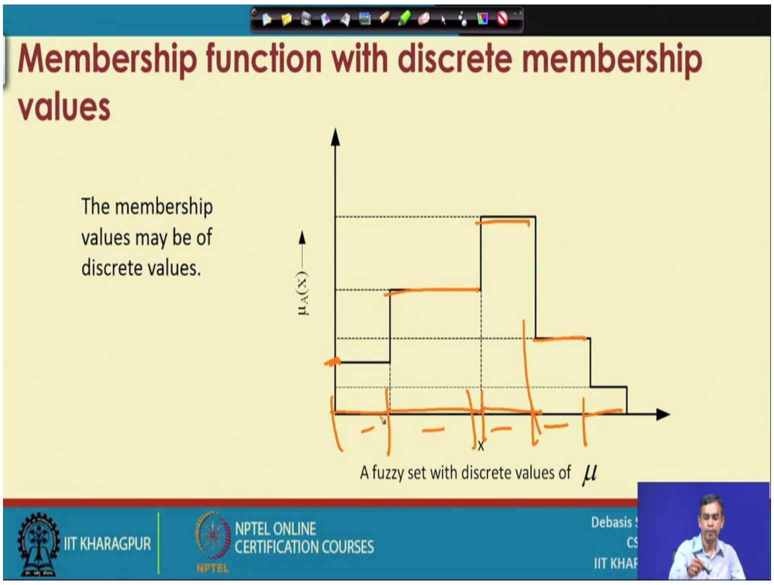
(Refer Slide Time: 27:05)



So, here as an example that how fuzzy set can be X is the all cities in India and A is a fuzzy set City of comfort and then this fuzzy set can be discussed using this form.

Now, membership functions may have any value. They are with either discrete membership values. Here I can show I show one example here where the all.

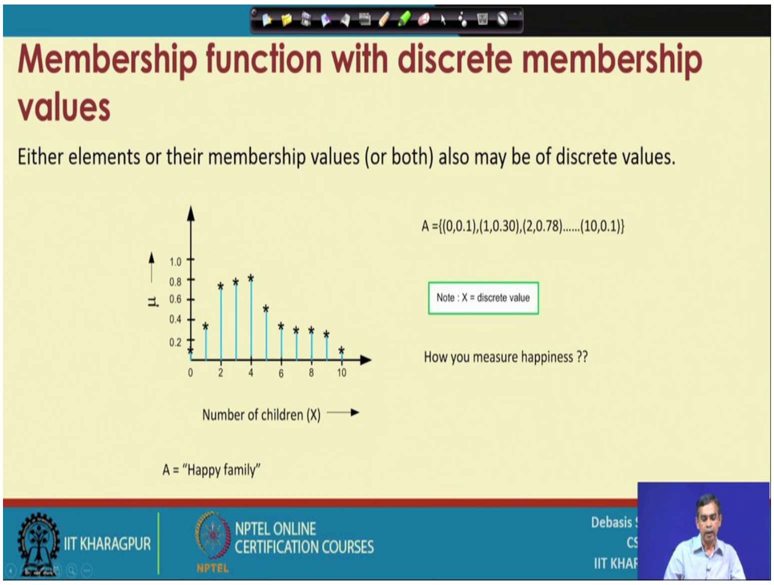
(Refer Slide Time: 27:22)



So, these are the elements in between has the membership value this one in these element the membership value is this one here these one so this one. So, the different elements so different element have the different membership value and it is called the discrete what is called the values of the membership function.

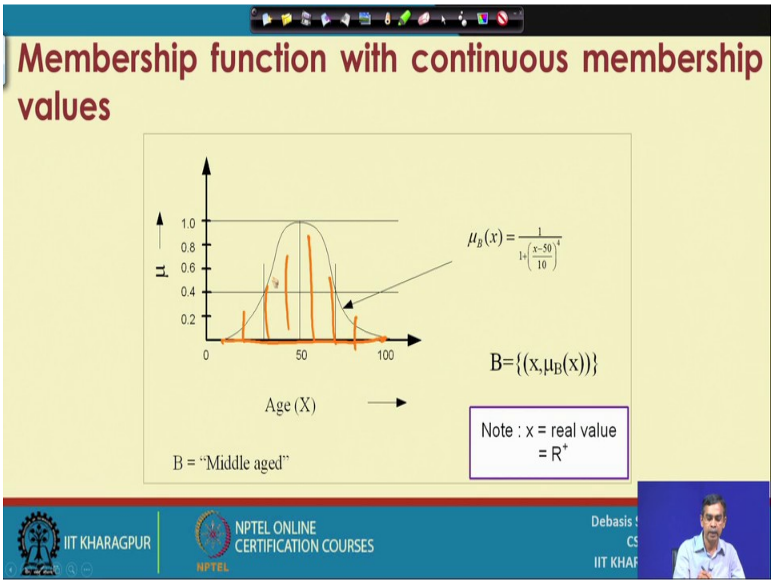
So, the membership values can be discrete, the membership values can be also continuous domain.

(Refer Slide Time: 28:00)



And the element also can be a discrete. Here in this example all the element that belongs to these are defined in terms of discrete, quantities. The membership values also may be discrete or continuous.

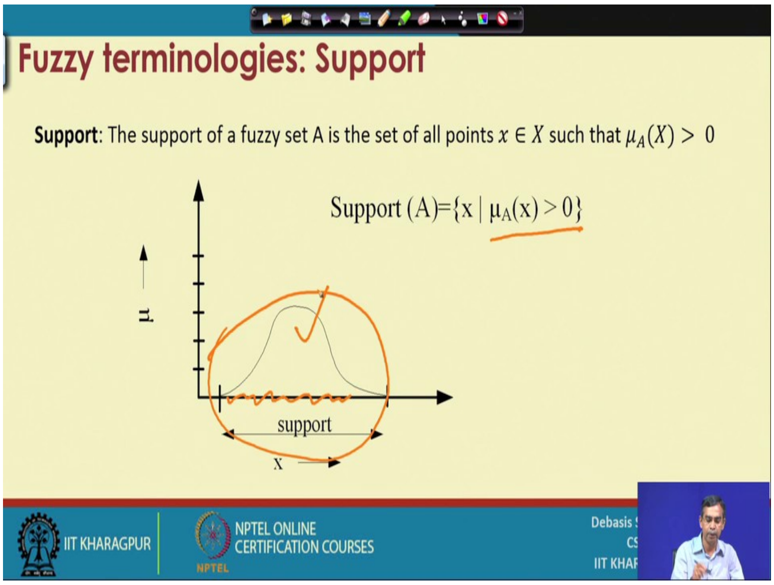
(Refer Slide Time: 28:27)



So, what I want to say is that the element can be either discrete value or continuous value likewise any membership values for element can be of discrete values or it can be of continuous values. So, this is an example which basically shows how the membership values is continuous. For example, in this region, so membership values for any element in a continuous domain can be described by means of this curve. So, it like this whatever it is.

So, membership value can be a discrete value element can be discrete value, the membership the value can be continuous the elements also can be continuous.

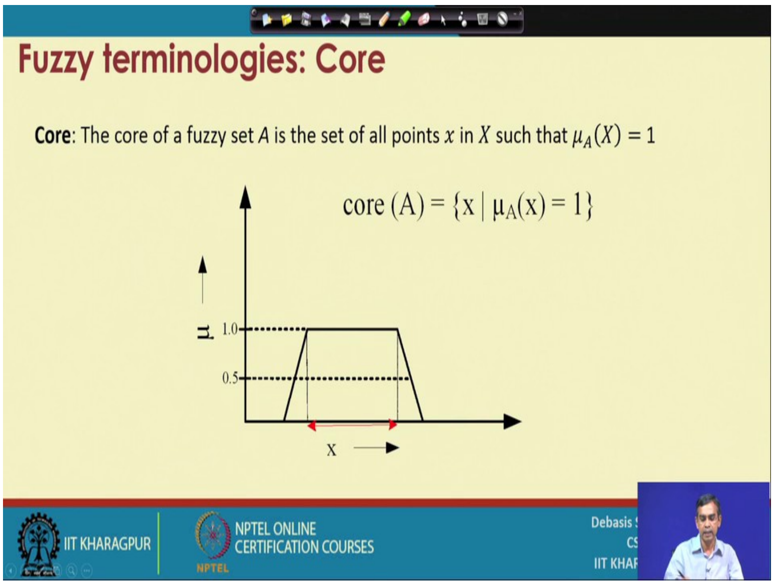
(Refer Slide Time: 28:57)



Now, there are a few more things or their terminologies are there. I will quickly cover this terminologies within one minute. So that we can understand about it. So, the first terminology is called the support. The support of an element is of a fuzzy set denotes that whose membership value is greater than x.

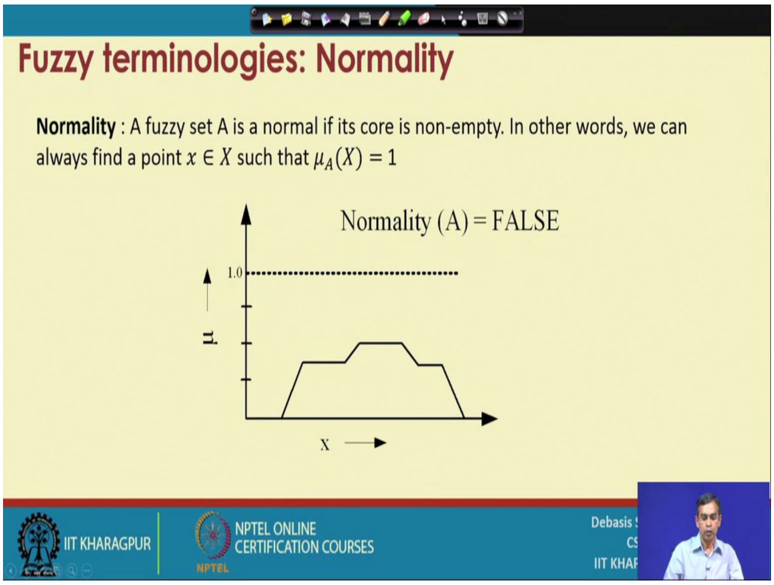
So, all these elements are basically the support which is belong to define this fuzzy set whose membership function is like this. So, what you can say that a fuzzy set in fact can be disclaim by means of a graph.

(Refer Slide Time: 29:26)



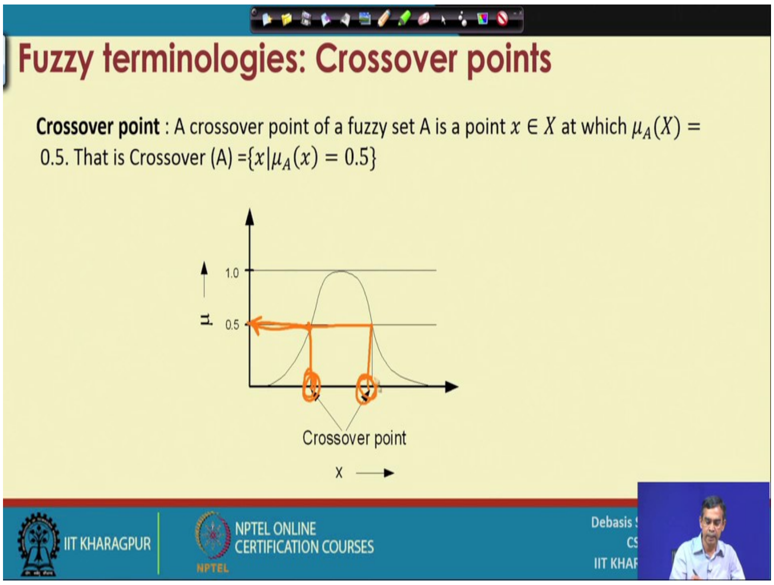
Regarding these things will discuss in detail later on. Now here core A. Core A is basically all elements a which are having membership values is equal to 1. Now here the core A all these elements having the membership values 1. So, these basically denotes the core A and we can understand that core A essentially a fuzzy sets.

(Refer Slide Time: 29:48)



So, normality. Now a fuzzy set can be a termed as a normal it is basically a Boolean value either 0 or 1 or the crisp value if it contains at least one element which core value is non-empty; that means, it has at least one element whose membership value is one. And if it does not contain any element whose membership value is not equal to one then it is not a normal. So, normality is false.

(Refer Slide Time: 30:15)



Now, crossover point. So they are the elements whose membership value exactly 0.5 is called the crossover point. For example, in this graph we can say this is the two elements it has the membership value 0.5 this also has the membership value 0.5. So, this element and this element whose belongs to the set x is basically the crossover point in this case. Few more terminologies we will discuss as the time is short so will discuss in the next lectures.

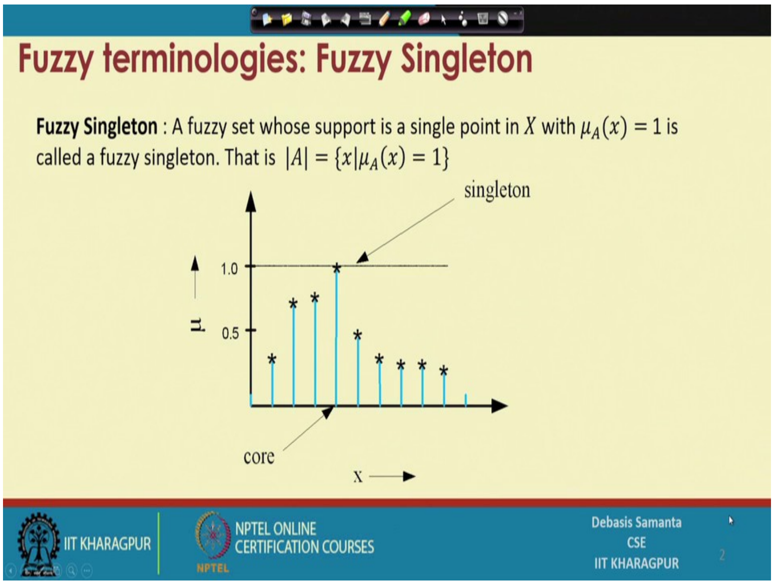
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 03
Fuzzy membership functions (Contd.) and Defining Membership functions

So, we are discussing about some notations and terminologies that is required to understand the concept of fuzzy logic. So, few terminology we have discussed in the last lecture and today we will continue the same discussion, we will discuss few more terminology and so first is called the fuzzy singleton.

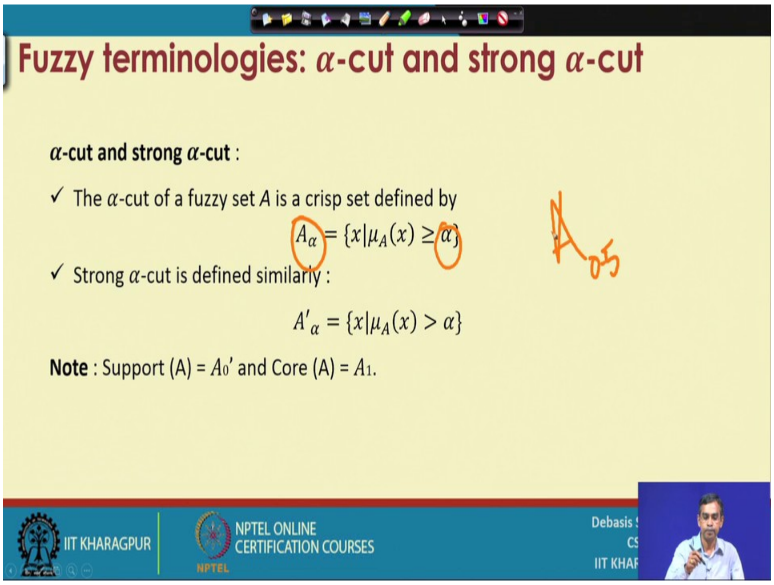
(Refer Slide Time: 00:30)



So, if a fuzzy set consists of only one element whose membership value is exactly one then such a fuzzy set is called the fuzzy singleton. For example, here this is the one fuzzy sets all the elements having the different membership values, but there is one element whose membership value is this one is basically one then it is called the fuzzy singleton.

So, fuzzy singleton is like this. And now, we will discuss about another two important terms it is called the alpha cut and then strong alpha cut.

(Refer Slide Time: 01:05)

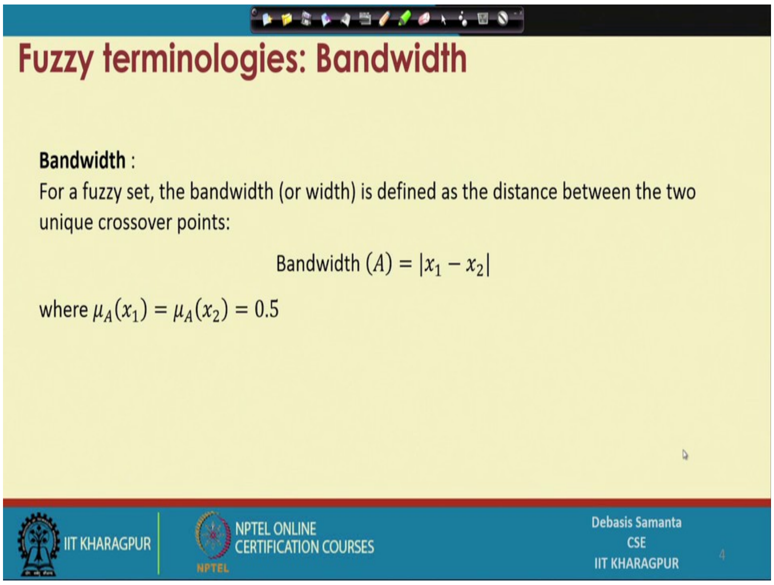


The alpha cut of a fuzzy set it is denoted as A_α, A suffix alpha it is denoted as alpha A_α. The alpha cut is basically the crisp set x set of element say x such that, the membership values of this element is $>=\alpha$, where alpha is a predefined values and need not to say that alpha is basically a value in between 0 and 1 both inclusive.

So, for example, $A_{0.5}$ if I say like this; that means, it is basically the crossover points, the set of all crossover points that belongs to the set A. Likewise the strong alpha cut the difference between the two is basically greater than equals and in case of strong alpha cut it is basically greater than symbol otherwise the they are the same.

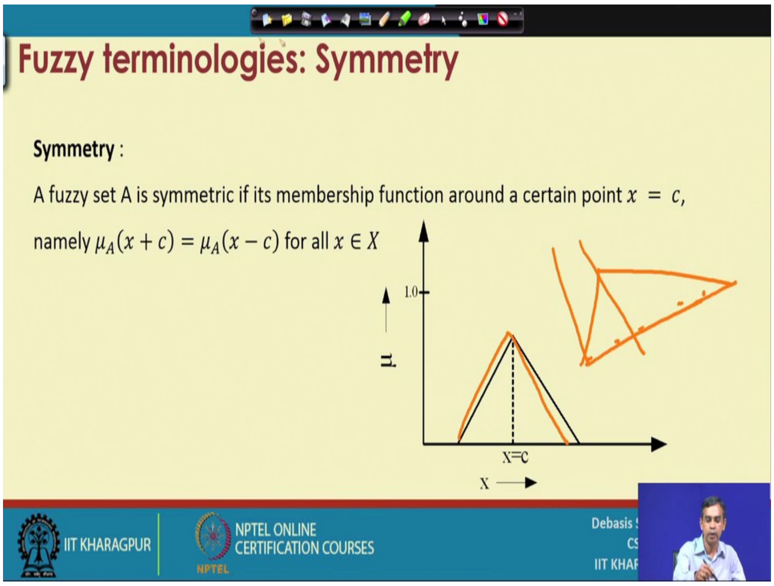
So, we can easily understand that a support that we have discussed about is at A_0 complements is basically complements and complement means other than the elements which belongs to 0 is A complement will discuss about the complement is just like a set complement you know inverse actually and similarly we also can say core A same as A_1 from the previous discussion that we know. So, it is like this. So, core A is basically the alpha cut where $\alpha=1$.

(Refer Slide Time: 02:49)



Now, we can define another term is called the bandwidth, bandwidth of a fuzzy set A. It is basically the difference between the two values of the element namely x_1 and x_2 such that x_1 and x_2 both are the two crossover points. Obviously, if there is if there is a fuzzy set which contains more than two elements at the crossover point then that two extreme crossover point can be used to decide their bandwidth. So, bandwidth is basically the difference between the two extreme crossover points x_1 and x_2 like this.

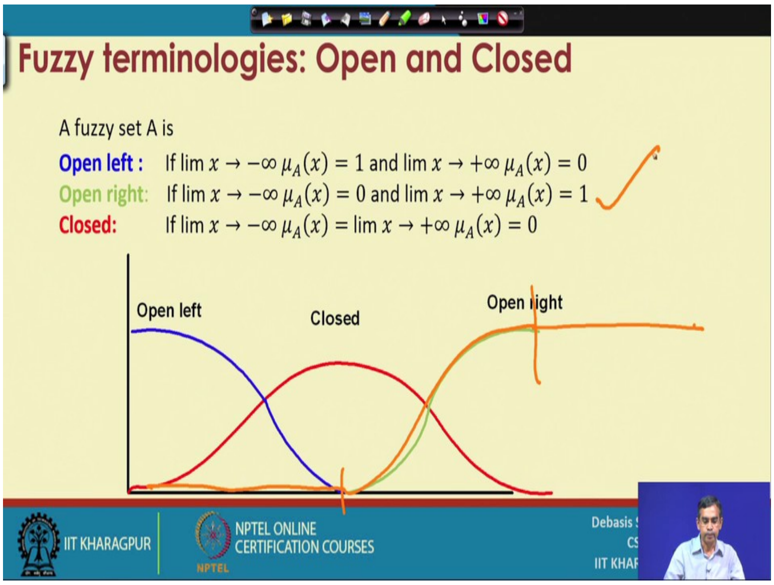
(Refer Slide Time: 03:37)



Now, we will discuss again a fuzzy set as a symmetric or asymmetric. We define a fuzzy set as symmetric with respect to one element $x=c$ such that all membership values for all the elements in this region has the same and there is a corresponding membership corresponding elements having the same values. Alternatively or mathematically we can say that if the two elements $x+c$ and $x-c$ have the same values for all element x that belongs to this set then we can say such a fuzzy set as the crisp set.

In other word, crisp set is basically symmetry in form. But if we draw one then this then this is not a crisp set. Because here some elements and all elements may not have the same values like this one. So, this is the concept of symmetric fuzzy set.

(Refer Slide Time: 04:39)

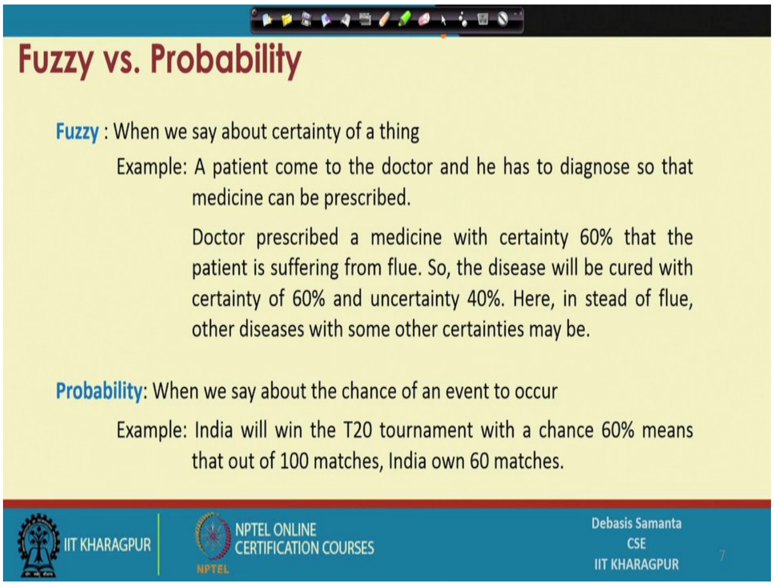


The next is basically open and closed fuzzy sets. Now, here this is the one description of fuzzy set we can say open left. Now, we say the open left a fuzzy set is if it satisfy this definition. Now, we can say for the open left all elements which is beyond the $x-\infty$ this side is =1 because it is basically one and limit $x+\infty$ after this point is basically 0. So, there are two extreme limits this one where all elements is 0 and here all elements is one then such a fuzzy set along within this portion is called open.

Similarly, we can write open right here, here all elements which $x+\infty$ is basically one and here all elements which $x-\infty$ is 0. So, this definition it is called the open right. So, open left and open right. Similarly the closed if we say for all element $x-\infty$ and all element $x+\infty$ if there the value is 0 then all this is basically called a fuzzy sets and this type of fuzzy set is called the closed. So, there are maybe three different form of a fuzzy set are there open left or open right or closed. So, any fuzzy sets can be belongs to this category only either open left, open right or closed.

Now, one thing just we want to clarify it is here and is there any link between fuzzy and probability. Now, there may be certain what is called the link or relation because if we see the fuzzy membership values is in between 0 and 1 both inclusive.

(Refer Slide Time: 06:49)



Similarly, if we say the probability value of something it also has the value in between 0 and 1 like this one.

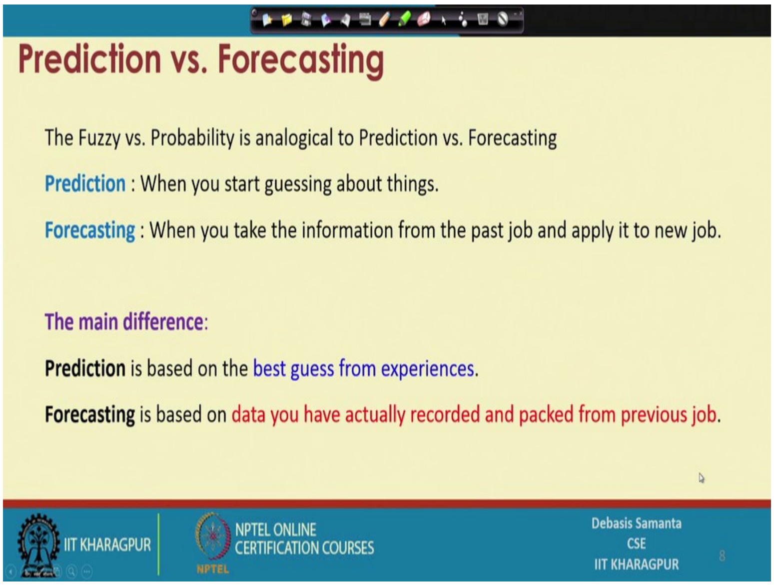
So, for the values are concerned both fuzzy and probability is synonymous, but there is a difference again. All these 0.1 or some decimal in between 0 and 1 alternatively can be expressed in the form of a percentage. So, 60 percent is equal to 0.6 like this one.

So, anyway, that whether it is expressing the form of a decimal 0 point something or it is a percentage there is basically relation between the two things in that way, but there is a clear cut difference between the two, I want to clarify this I mean difference between the two with an examples. So, first example suppose a patient when come to the doctor and doctor carefully diagnose the patients and prescribe the medicine. Then what exactly the thing it happens is that doctor prescribes a medicine with certain certainty, let this certainty be 60 percent. This means that the patient is suffering from the flue and for that he is sure about this is that 60 percent. In other words, that disease for which he prescribed the medicine will be cured with certainty of 60 percent and there is again uncertainty 40 percent. So, this is a concept that is related to the fuzzy actually it is the certainty or clarity or the guarantee like.

On the other hand if it is the probability then we can say that probability is also 60 percent or sort of things or 0.6 like.

For example, India suppose we will win the T-20 tournament with a chance 60 percent. If I say so it means that, we have certain statistics or some previous experience that out of hundred matches India won 60 matches. So, 60 percent in this case and 60 percent in the previous case has the two different significance. So, in the first case in case of doctor patient scenario it is basically the certainty and in the second case the 60 percent is based on the previous experience. So, this certainty versus experience this basically defined the fuzzy versus a probability.

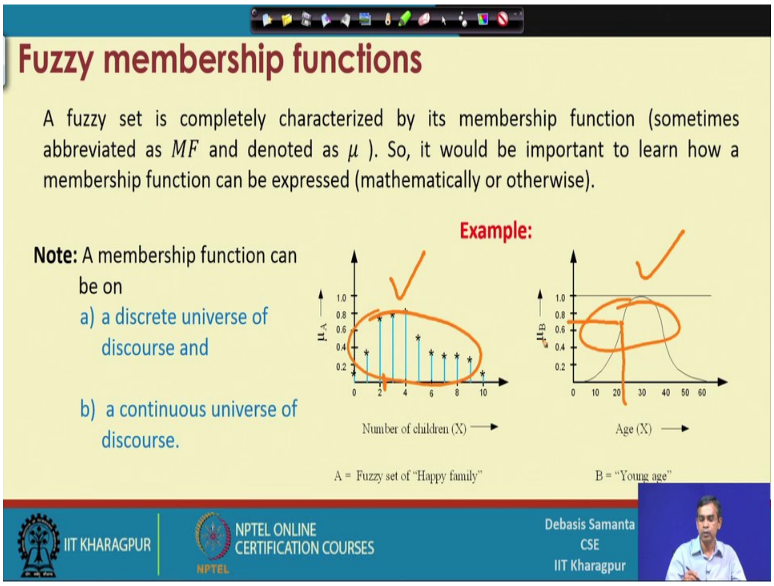
(Refer Slide Time: 09:12)



Now, likewise this fuzzy versus probability there is another relation that is also related to this fuzzy and probability also it is just in the form of prediction versus product forecasting. So, fuzzy versus probability is also in many way analogical to prediction versus forecasting.

Now, we can say the prediction when you start guessing about something. So, it is a guess, fuzzy means it is a guessing power. On the other hand forecasting means if we can say something based on the previous information or based on our previous experience, it is the forecasting. So prediction, in other words, the prediction is based on the best guesses from the experts that basically who does it and forecasting is based on the data which you are already have in your mind and based on the processing of the data you can you can tell something. So, this is a forecasting. So, if prediction is related to fuzzy then we can say the forecasting is related to the other, that means, its probability. So, these are the things actually. So, sometimes we little bit get confused what about the fuzzy versus probability or like this one.

(Refer Slide Time: 10:28)

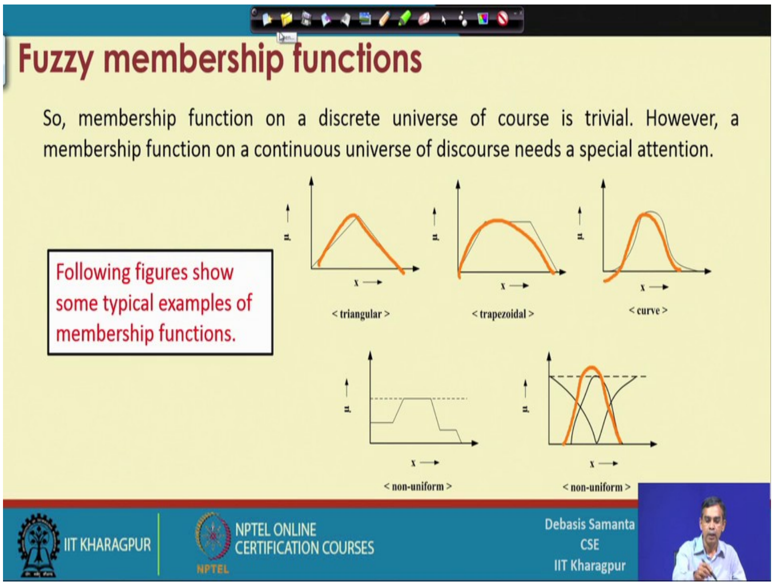


Now, next our point of discussion is basically fuzzy membership function. We have some idea about it and one thing I want to again mention it here a fuzzy set can be described better in the form of a graph that mean it is a graph versus all elements and their membership values. So, we can define in the form of a set theoretic form or in the form of a graph. So, in this slides we can see this is the one fuzzy sets and this is also another fuzzy sets.

Now, the difference between the two fuzzy sets here is that here the elements which belongs to these fuzzy sets are having discrete values. So, there will be no element which in between 2 and 3 like. So, there will be no element there. But here is all elements in between the range 0 to 60 are belongs to this one. So, it is the elements say 21 and it is the fuzzy set is there.

We have also discussed about that the membership values can be discrete value also. So, here it looks at all values are possible. So, here also the continuous values for all membership values and here need not to say it is also the continuous values of all membership values. So, that we have already discussed in the previous lectures that the membership function can be on a discrete universe of discourse or can be a continuous universe of discourse, the membership value can be again discrete values or it can be again continuous values. So, whatever be the values it is there we have to express, maybe it is mathematically or using some graphical representation. So, these are the two examples where we had give the two fuzzy sets in the form of a graph.

(Refer Slide Time: 12:17)

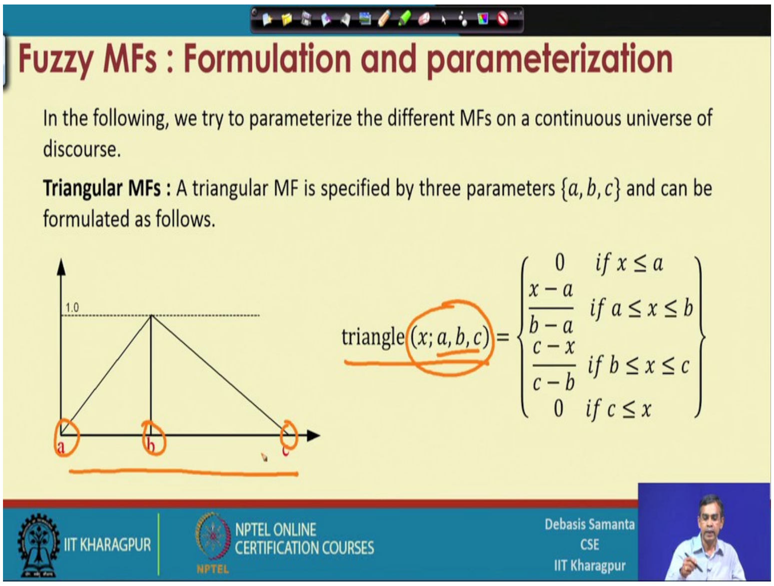


Now, I want to give more graphical representation of the standard some fuzzy sets. So, these figures basically show some typical examples of the fuzzy sets and they are defined in terms of the membership function actually. And that this is the general looks that usually a fuzzy sets takes. And in the first, this is the one fuzzy set it is having the membership function in the form of a triangular shape.

So, this is another fuzzy sets whose membership function is expressed in the form of a trapezoidal shape. This is another membership function I did basically curve it is look like a bell curve. So, it is called bell function bell membership function like. And it is the one membership function which does not have any a specific shape it is arbitrary shape is called the non uniform fuzzy sets. And this is also an example it is also a non-uniform fuzzy sets, however, it has some special meaning. So, this is the one fuzzy set it is called the open left, this is another fuzzy set is called the open right and this basically called the fuzzy set closed. In this sense it is also a closed, these are all the closed fuzzy set actually.

So, these are the all closed or open whatever I told you that either fuzzy sets can be open or open left, open right or is a closed anyway. So, these are the typical form of the fuzzy set which usually consider in our fuzzy system, in our fuzzy theorem and another point is that how such a fuzzy set can be better described in some mathematical notation so that we can process them in our future fuzzy system design.

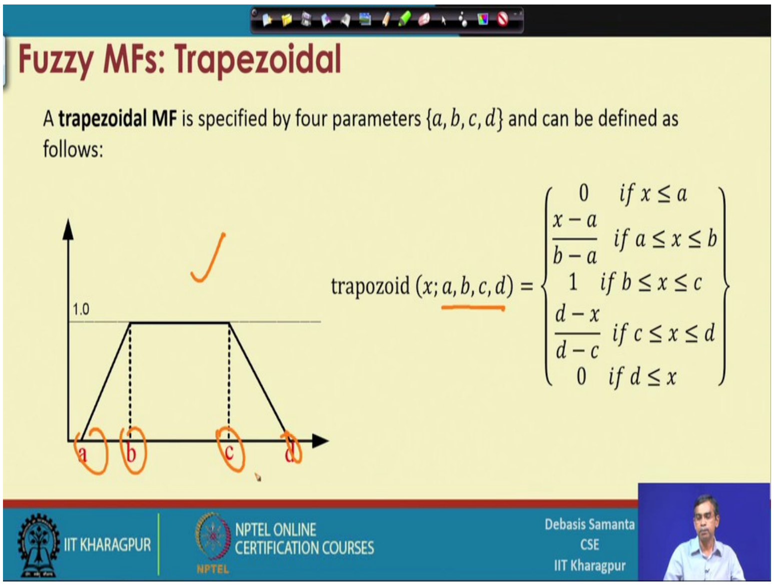
(Refer Slide Time: 14:02)



I am going to discuss these things how a membership functions can be better mathematically described and then that mathematical specification can be used to process in subsequent requirement. So, in this direction first let us discuss about this is the one fuzzy sets or a membership function and this membership function as I told you it is a triangular membership function. So, usually this is can be described mathematically using this triangle and x is an element and this membership function can be defined by means of three parameters a b c. So, three parameters are the three what is called a meaningful direction here and in terms of these three parameters the membership function can be described mathematically using like this. So, it is clear that if $x \leq a$, it is like this then this is 0 and in between a and b the membership function can be defined by this, this is basically slope whatever it is.

Similarly, in between this one, this is another slope it can be defined like this and so for the element $x > 0$ this one it is basically 0. So, what you can say whatever the graphical representation which looks like this it can be described mathematically using this form. So, this is a mathematical expression that a fuzzy set can be defined and definitely this fuzzy set is defined over a continuous universe of discourse.

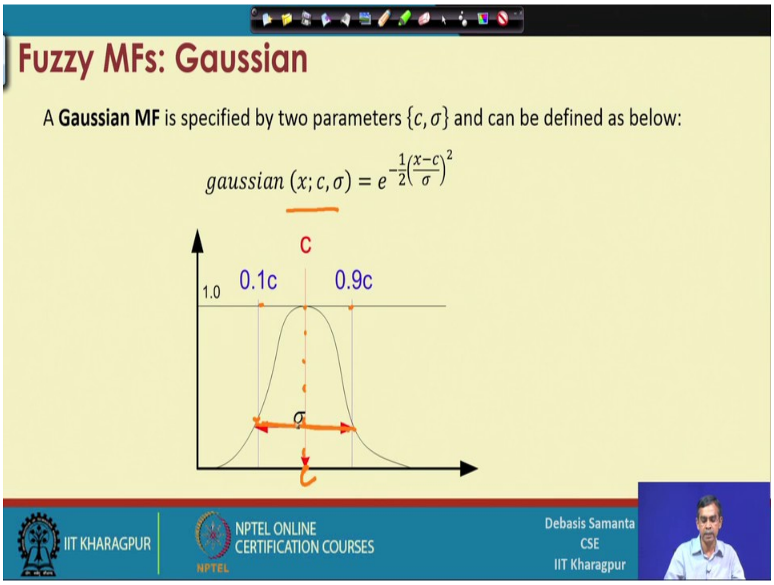
(Refer Slide Time: 15:48)



Now, this is the triangular membership function using the same concept same idea we can define the other membership function. For example, in this slides we can show the trapezoid the membership function. However, unlike triangular membership function here we need the 4 different parameters they are called a b c d. So, these are the parameters are like here and in terms of these 4 parameters we can define this membership function like the if $x \leq a$ it is 0. And if x is in between a and b that means, in this one, so it is basically this one, which can be expressed by this form. And in between b and c this is a x, so it is basically 1. And in between c and d this is the slope which can be described by this one and if $x > d$ then this is 0.

So, with this, this concept can be describes in a mathematical manner this membership function now. So, this is a trapezoidal membership function triangular and trapezoidal are the two most frequently used membership function in fuzzy system in order to design a fuzzy system.

(Refer Slide Time: 16:55)

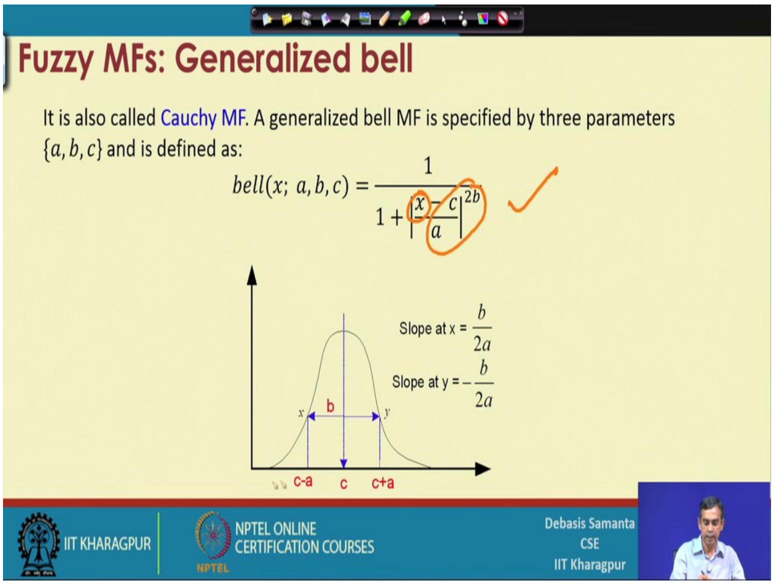


The another important membership function that is more popular in fuzzy in order to describe a fuzzy system and this is called the Gaussian membership function.

A Gaussian membership function typically takes in terms of two parameters c and sigma. So, is c basically is the middle point of this it is called the centroid or median mean and sigma is defined is a range between these two in between 0.1c and 0.9c if c is defined here then this 0.1c and 0.9c. So, this range is basically sigma.

Anyway, so if we define two parameters like c and sigma then this membership function can be better expressed in the form of a mathematic notation using this one. So this basically the formula for Gaussian distribution, that is why it is called the Gaussian form. If we plot this form for a given values of c and sigma and for different values of x then the graph will look like this. As this graph is looks like a bell shaped, so it is called the bell shape membership function also.

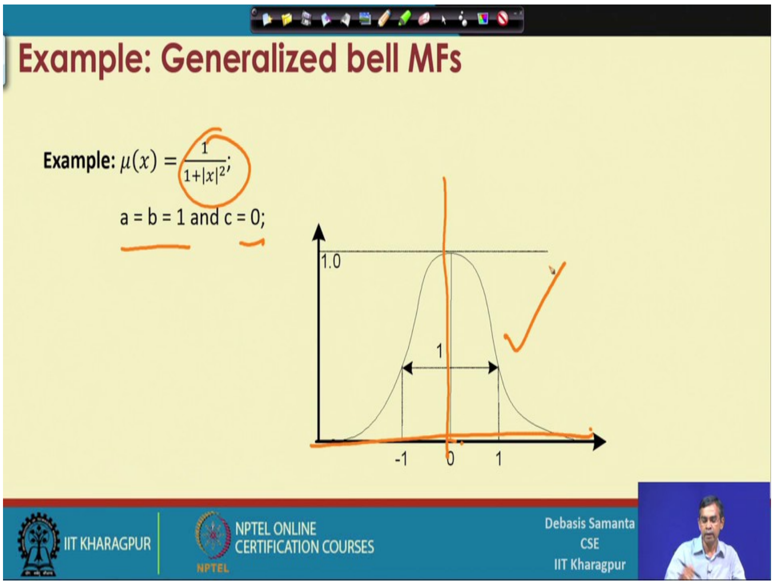
(Refer Slide Time: 18:12)



So, this is the one popular membership function like this is the membership and there are many more and another is called the Cauchy membership function. So, it is just like a Gaussian membership function, but this membership function defined in terms of three parameters a b c. And using these three parameter how a membership function is defined it is expressed here and it has two characteristics here that it considered two points for the slope. So, slope at this point it is called the $c-a$ and another is $c+a$ the slope of this point is basically $b/2a$ and at this point is $-b/2a$.

So, if we define a b and c then all these values their slope and point can be defined and accordingly the membership function can be defined. So, the membership function that can be defined using Cauchy membership function is this and if we plot this membership function for a given the values of a b c then the graph will look like which is shown here.

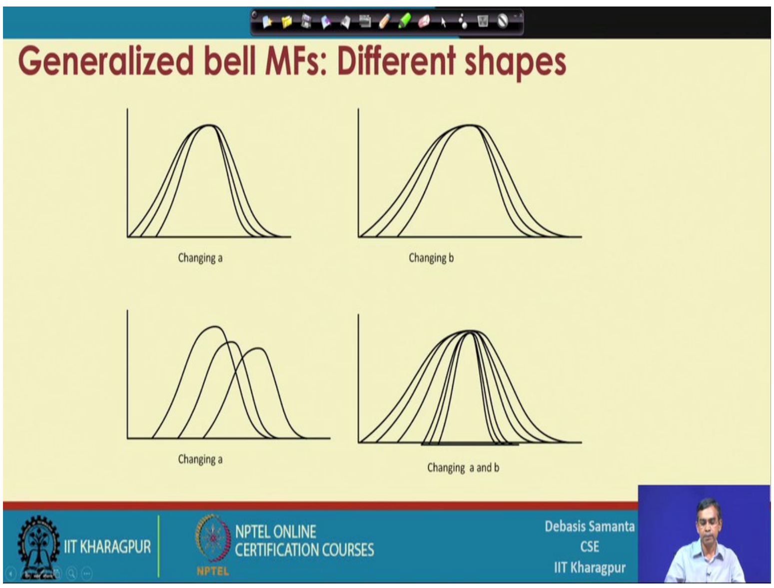
(Refer Slide Time: 19:14)



So, this is the another popular Cauchy membership, Cauchy membership function or it is sometimes is called the generalized bell.

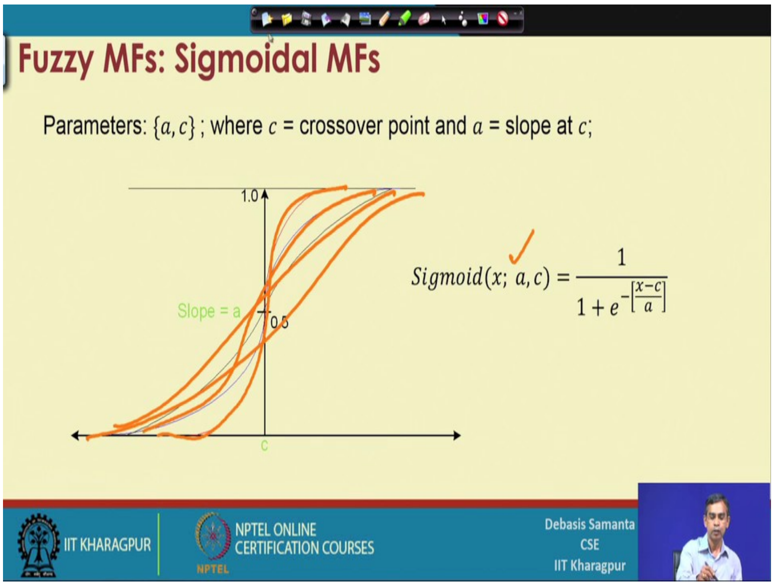
Now, this is a typical example of generalized bell for particular value of a b c where $a=b=1$ and $c=0$; that means, it is basically $c=0$ and these are the function and if we plot these things you essentially if we plot this one then it can give back up like this. So, we can plot it and then you can have this curve. So, this is basically graphical representation of this one.

(Refer Slide Time: 19:47)



So, there are few more membership function. Anyway before going to these things for the for the different values of a b c and whatever it is there, so this membership function will look the difference shape and hence the different membership function or the different fuzzy elements can be defined. So, this is basically the idea about the membership function can be expressed in the form of a graph as well as in the form of a some mathematical notation.

(Refer Slide Time: 20:06)

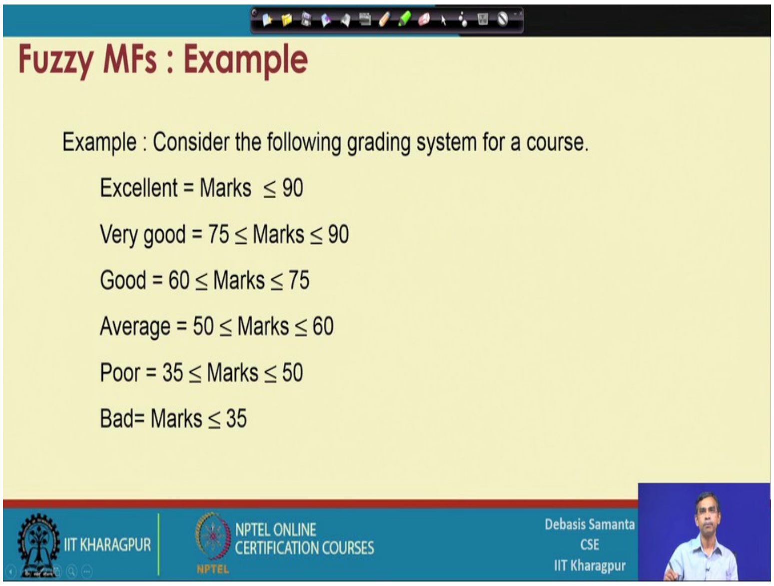


Now, this is another very popular the membership function it is called the sigmoidal membership function. A typical look of the sigmoidal function is shown here in this form. It is basically indeed form the s. So, that is why it is called the sigmoidal membership function just like a s. This kind of the membership function defined in terms of two parameters a and c, where c denotes the point it is like this and a is any arbitrary value, it is basically called the slope of this point at a.

Now, if we take this kind of values then the sigmoid function can be plotted and if we follow this expression. So, if this is the expression if we follow then graph can be obtained for this expression for different values of a, it is shown like here. So, we can see here one example. If $x \infty$ as you show it is basically, in this way is basically closed right and in this case it is basically open right. Because for all value the x which is ¿∞ the membership value is 1 and few here for all value the $x - \infty$ the membership value is 0.

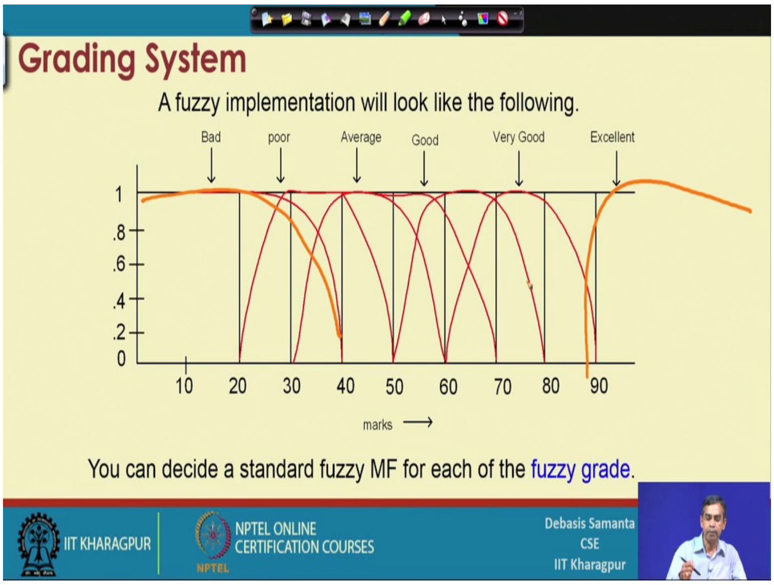
So, sigmoid is typically like this and for the different values of a, the different curve will be obtained. For one values of a, the curve will be like, for another value of a the curve will be like, for another is there and so on. So, if we change the values of a, the different pattern of the curve will be obtained. So, what I want to mention here is that this is an important function, which can have the different values of a and c and the different membership functions can be obtained and hence the different fuzzy sets, the different form of the fuzzy set look of the fuzzy set also can be obtained. So, this is another membership function that is very much popular in the design of fuzzy system.

(Refer Slide Time: 22:08)



Now, we come we can discuss one example. So, that how these membership functions can be used. We are discussing about one idea about that how the grading in the form of a crisp.

(Refer Slide Time: 22:21)



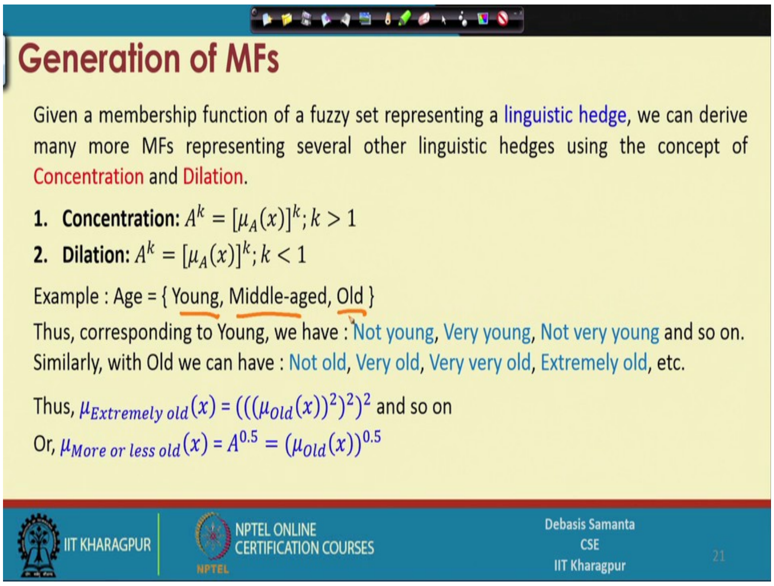
So, these are the crisp formulation of the grading and here the same thing which can be discussed in the form of a fuzzy, another fuzzy representation. So, for example, the grade this one is one and grade is this one. So, these are the two different form. So, as we say, this can be discussed with some function and this kind of membership function can be discussed by means of say Cauchy MF or generalized bell like. So, basically this membership function can be used to different what is called the fuzzy sets in the different range. So, this is a one fuzzy set in one range, this is another fuzzy sets and this is another fuzzy sets like.

So, we can use the same formulation for defining membership function. In this case other than these sets or other than these sets, these two sets can be defined either using open left or closed left with another membership function, but all the other membership function in between the range and they can be defined in terms of some graph or in terms of some mathematical notation say Cauchy membership function like and then they are basically called the different fuzzy set. For example, if we define one membership function in this form, we can say that these are bad fuzzy sets; where the universe of discourse is this one; and the elements belong to this is this one; within this element up to this is the membership elements see value is one and after this thing membership value is decided by this what is called the function.

Similarly, the another membership function like this one. It can be defined in terms of this is a universe of discourse; the elements is in between, for all elements it is 0, for all other element is 0 and in between this the membership value will be decided by the type of the curve. So, these are the basically the way which we can use to define the membership function for the different elements.

So, we have understood about the different membership functions and their mathematical representation.

(Refer Slide Time: 24:24)



We will quickly cover few more concept about the membership function the two concept it is called the concentration and the dilation. So, the idea is that, if A is a fuzzy set given to you right, then we can define another fuzzy set then let this fuzzy set be A^k, where k is some value that may be greater than equals to greater than 1, such that this [$\mu_A(x)$ ¿¿^{k} is the new membership values for the element x which is belongs to the set A^k. So, this means that if A is known with $\mu_A(x)$ is a membership value then we can derive another fuzzy set A^k with membership value this one. And for k > 1, if it is like this, then it is called the concentration.

Similarly, if it is k < 1 the same thing if whole goods then it is called the dilation. So, this is a very important concept that from a given fuzzy set we can find more fuzzy sets many more fuzzy sets, this fuzzy sets can be obtained by simply using some mathematical notation like this concentration or dilation.

Now, you can recall each fuzzy set basically used to express something say high temperature or low pressure or sweet apple whatever it is there. So, such a things in fuzzy concept it is called the linguistic hedge. That means, high temperature the linguist hedge it is that temperature is high, but temperature is high; that means, the different temperature is high with the different membership values.

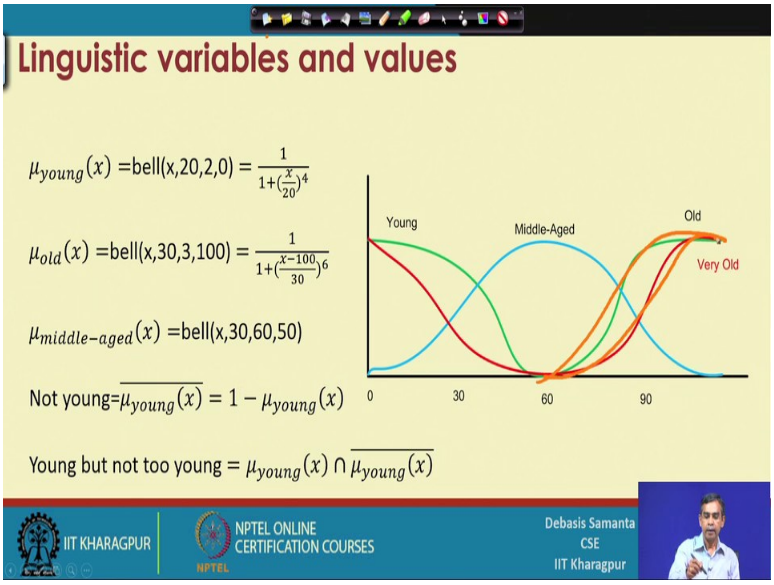
Similarly, if we say the pressure is low; that means, the different pressure can be termed as low and then same pressure can be termed as belongs to low pressure or high pressure. So, here pressure low, pressure high they are called the linguistic hedge. Likewise there are many linguistic hedge. For example, related to our age. We can define related to our age say young, middle aged and old. So, these are the three fuzzy sets if you can define then from these three fuzzy sets, we can easily define another fuzzy set likes a very young, not very young or like this one. Similarly if the old is a fuzzy set, very old, very very old, extremely old all these fuzzy sets.

Now, the question is that how these kind of fuzzy sets can be obtained if a fuzzy set is available already with us. Here is an example we can use or you can use the concept of concentration and dilation to do these things. Now, here for example, $\mu_{Extremely\,old}$ say if we know that μ_x is a fuzzy set which is defined for the old fuzzy set and x is any elements belong to the fuzzy set old then any elements belongs to the fuzzy set extremely old having the membership value $\mu_{Extremely\,old}(x)$ that can be obtain like this. So, it is basically $\mu_{old}(x)$

that is the original fuzzy sets having the membership value and we can take $\begin{matrix} x \\ (¿¿2) \\ (¿¿2)^2 \\ ¿ \\ ¿ \end{matrix}$. So, this is basically gives that very old, this is very very old and this is basically extremely old. So, these are the different linguistics can be obtained and here we have used the concept of concentration.

Now, similarly if more or less old if μ_x is defined for the old fuzzy sets then more or less for k value say 0.5 we can define this one. So, it is called the dilation. Now, graphically the same thing can be plotted nicely we can say it. Here is an example.

(Refer Slide Time: 28:04)



So, we can write it. So, if suppose this is the fuzzy set young. Then by applying the dilation we can say the very young we can define like this another fuzzy set. Similarly if this is the fuzzy set old then using the concentration we can define another fuzzy set a very old like this one.

So, the different fuzzy sets can be obtained different fuzzy sets can be obtained from existing fuzzy sets if we follow some concentration and dilation formula. Here is few example again, $\mu_{young}(x)$ is one fuzzy set defining membership function fuzzy set young defining each membership function as $\mu_{young}(x)$ and it is supposed defined by means of bell shaped curve that we have discussed and this is the curve. Then μ_{old} also another fuzzy set which is defined by another bell shaped curve which is like this one.

Now, given these two we can define the other not young. So, not young can be defined $1-\mu_{young}(x)$. So, this is the another fuzzy sets the value or membership values belongs to another fuzzy set let the name of the fuzzy set be not young. So, this can be derived from the fuzzy set young. So, as another example young, but not too young we can define this kind of concept. So, $\mu_{young}(x)$ and it is complement of this one. So, it is young and it is not young. So, young, but not young can be defined by this one.

Now, regarding this kind of formulation we will discuss in details in our next lectures. So, what I want to say here that given a fuzzy set we can derive another fuzzy set easily using some mathematical computation and formulation. So, these are the things it is there we have discussed about the concept of fuzzy sets first and then we learned about what is the difference between crisp set, versus fuzzy sets. Subsequently we learned about the different membership function that is with which we can define fuzzy sets and then different properties in it. And how the membership function can be better mathematically expressed also we have learned it. And we will discuss about the different fuzzy set operation in our next lecture.

Thank you.

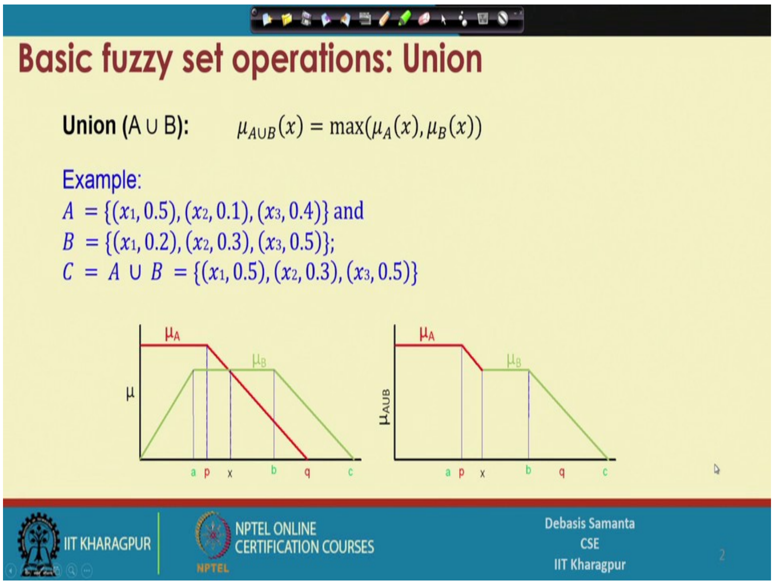
Introduction to Soft Computing

Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 04
Fuzzy operations

So, we have learned about fuzzy set. Fuzzy set is the basic element in fuzzy system. Today we are going to learn about the different operations on fuzzy sets.

(Refer Slide Time: 00:30)



So, the different operations on the fuzzy set in many ways similar to the operation those are application to the crisp set. And this is the normal set that we have learned in set theory, the operation those are applicable they are in crisp set such as union, intersection, complement all those things also are applicable to the fuzzy sets. But the definition of all these operations are with different implications. So, we learn one by one the operations on the fuzzy set.

So, let us first discuss about the union operation of two fuzzy set. Now here A and B are two fuzzy set suppose. So, union of two fuzzy sets is denoted by the symbol $\cup$. This is the usual symbol that is used they are in crisp set. So, $A \cup B$ denotes the union of two fuzzy sets A and B. Now the union operation whenever it is performed on the two fuzzy set it will basically give the membership functions for each elements which are belongs to the union of two fuzzy sets, the membership values for different elements in the union of the two fuzzy set is defined by this expression.

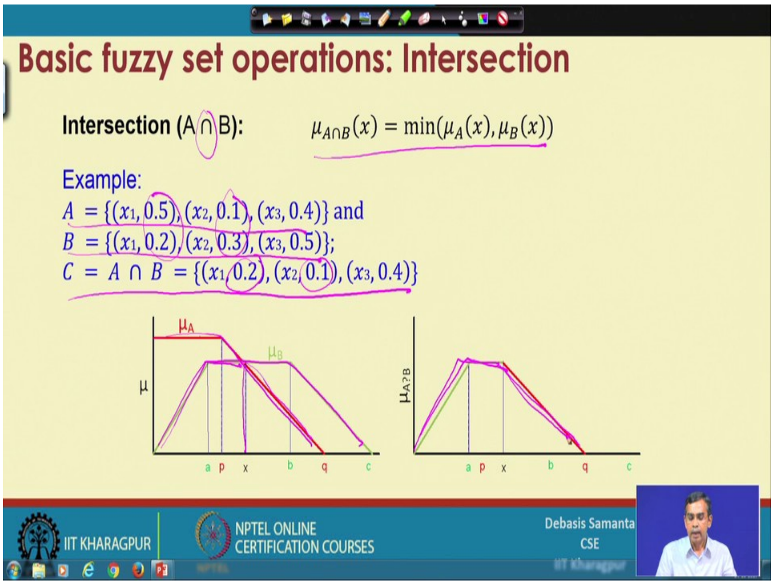
So, if we see this expression. So, union operation for any element x which belongs to the union of two fuzzy set A and B is basically $max(\mu_A(x), \mu_B(x))$, where $\mu_A(x)$ denotes the membership values for an element x belongs to fuzzy set A and $\mu_B(x)$ denotes the membership value of the any element x belongs to fuzzy set B.

Now, let us have an example. Say, here A is a fuzzy set which is defined with the different membership values for different element shown here, similarly B is the another fuzzy set. Now the union of two fuzzy set is denoted as C which is $A \cup B$ is basically this is the fuzzy sets. So, union operation gives another fuzzy set. Now here we can see how we have obtain the membership values for the different elements which belongs to the sets C. So, here 0.5 it is obtained as the max of these two value 0.5 and 0.2 it is as per the definition. Similarly 0.3 it is basically max of the two values membership values they are in the set A and B and likewise. So, this way we can obtain the union operation of the two fuzzy sets.

Now, the same thing can be better explained with the help of a graph. So, here we see the graphical representation of two fuzzy sets A and B. The graphical representation means it is basically a representation of the membership function how they change a change with the different values of elements belongs to the two sets. Say here the two sets with the universe of discourse. So, this is basically the universe of discourse of the two sets and the fuzzy set A is defined by it is membership function for the different element, which is denoted here, this is the membership element, membership function for the fuzzy set A. Similarly this denotes the membership elements for the fuzzy set B.

Now as the union operation is basically max of the two values. So, upto this portion so this is the B and this is A, so the max is this one. So, these basically the union upto this part upto this part and then so union of these and these basically the max. So, it is this 1 this part is the rest of the value upto this one. So, this way we obtain the membership function of the union of the 2 fuzzy set which is represented by this graph.

(Refer Slide Time: 05:02)

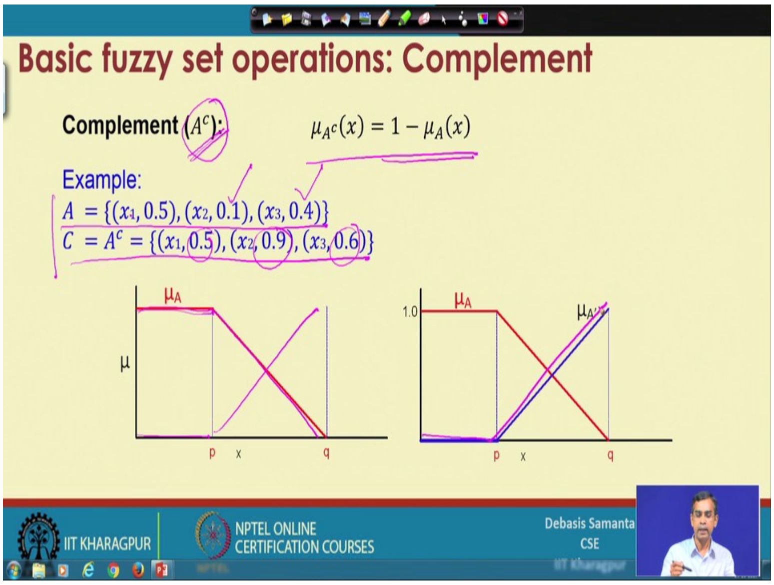


Now let us discuss the intersection operation. Intersection operation of two fuzzy sets is denoted by this symbol it is $\cap$. So, here and the membership functions of the resultant fuzzy set $A \cap B$ is denoted by this expression and you can say in case of union it was max whereas, in case of intersection we have to take the minimum of the two values of the membership for which the A and B belongs.

Now, as an example A is the another set and B is the another set and the intersection of the two set is represented here. Now here whenever we go for intersection for the element x_1 then we have to take the minimum. So, we take the minimum from here. So, it is the minimum. So, here 0.1 which is in A and 0.3 for x_2 which is in B and in the union operation where the x_2 has the value minimum this one so 0.1. So, this way we can obtain the intersection of two fuzzy sets.

Now, the same thing again can be drawn graphically. So, this is the fuzzy sets A and this is the fuzzy sets B and we have to take the minimum of the two. So, for upto these part minimum of A and B it is basically this one. So, we can obtain this part and for the rest minimum of this one is this one, so we can take this one. So, this graph basically shows the membership functions of the union of two fuzzy sets A and B.

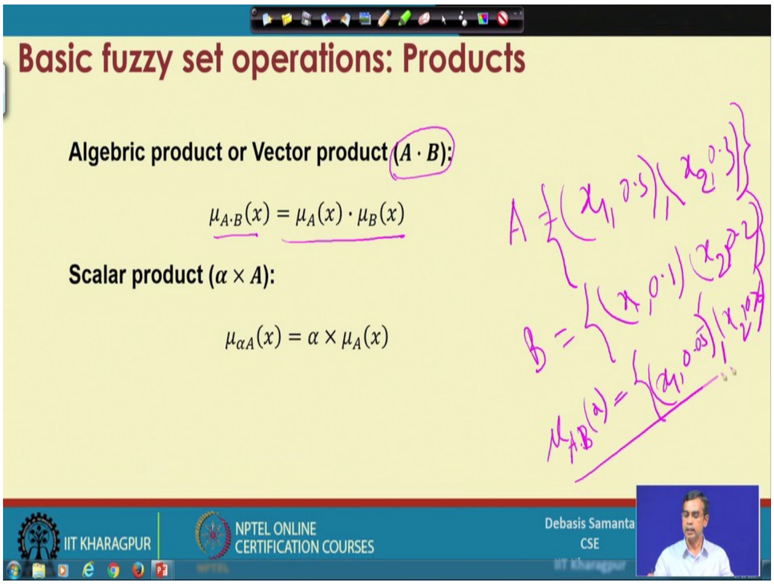
(Refer Slide Time: 07:01)



Now another operation on the fuzzy set is called the complement. The union operation and intersection operation are the binary operation, because they needs two fuzzy sets whereas, the complement operation is an unary operation, which is applicable to only one fuzzy set. The complement operation on any fuzzy set A is represented by the symbol A^c. So, it is basically a complement of the fuzzy set A and the resultant value of the membership functions is defined by this expression $\mu_{A^c}(x)$ is; that means, is a membership function of the resultant fuzzy set is c and it is denoted as $1-\mu_A(x)$ where $\mu_A(x)$ denotes the membership function for the fuzzy set A.

Now, if this is an example. So, A is the fuzzy set and then it is compliment A^c is like this. So, for x_1 we have to take the this formula $1-\mu(x)$, so it is 0.5. So, for the 0.1 it is 0.9 for 0.4 it is 0.6. So, the complement operation is straight forward. The same thing again can be shown graphically. So, this is the fuzzy set for this is the fuzzy set A where the membership function is like this. Now it’s complement, so the complement value of these is basically this one and complement value of these basically this one. So, the resultant value of the membership function for the complement A is basically this one. So, this way the complement operation can be obtained.

(Refer Slide Time: 08:55)

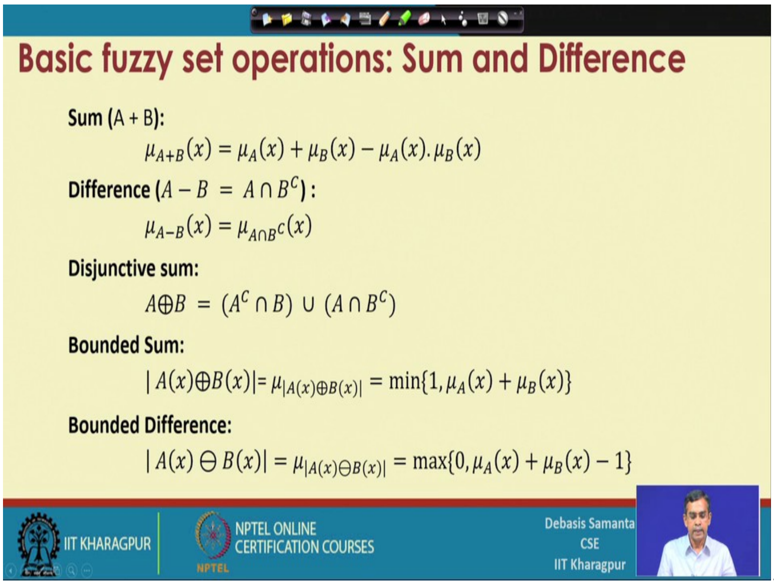


So, these are the very simple operations union, intersection and complement. There are few more operations. The first operation in this context is called algebraic product or vector product. Now algebraic product or vector product is denoted by a dot symbol; algebraic product or vector product is denoted by the dot symbol, $A \cdot B$ where A is a fuzzy set and B is another fuzzy set. So, the membership function for the resultant fuzzy set is denoted by this and it is obtained using this formula. It is basically simple as a product of the values of the membership functions belongs to the fuzzy set A and B.

Now if A is a fuzzy set $\{x_1, 0.5\}$ and $\{x_2, 0.3\}$ this is the fuzzy set and B $\{x_1, 0.1\}$ $\{x_2, 0.2\}$ then the $\mu_{A \cdot B}(x)$ for C that can be obtained. So, x_1 and it is basically product 0.5 and 0.1. So, it will give you 0.05. Similarly x_2 it will give you 0.06 and so on. So, this way the product can be obtained.

So, this is the vector product. Now like vector product is a scalar product where α is a constant; α is a constant usually value in between 0 and 1 both inclusive. So, the scalar product of a vector A is denoted by this formula, $\mu_{\alpha A}(x)$ where αA is a basically scalar product of the fuzzy set A and it's new membership value is defined as is a product of α and the membership values of the function $\mu_A(x)$.

(Refer Slide Time: 11:24)



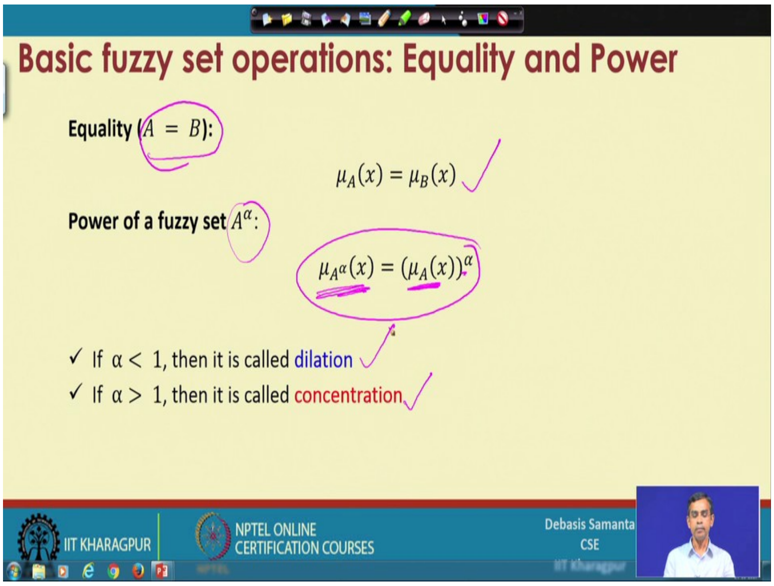
Now, there are few more operations one by one let us discuss them. So, the first operation is sum of the two fuzzy set A and B, it is denoted by the sum of the two fuzzy set A and B is denoted by $A+B$, + operation and it's new membership function is denoted by this expression. It is easy to evaluate if we know the $\mu_A(x)$ for the fuzzy set A and $\mu_B(x)$ for the fuzzy set B, then μ_{A+B} for the fuzzy set $A+B$ obtained using this formula.

Now, the difference operation on two fuzzy set A and B is denoted by this notation $A-B$, which is equivalent as the $A \cap B^C$ and it can be obtained using this expression μ_{A-B}; that means, membership value of the difference of the two fuzzy set A and B is same as the membership values of $\mu_{A \cap B^c}(x)$. So, if we can calculate. So, better idea is that we can calculate the $A \cap B^C$ and then the μ can be obtained easily from there.

Now, there are disjunctive sum it is denoted by this expression $A \oplus B$ like and it can be obtained using this one. So, $(A^C \cap B) \cup (A \cap B^C)$. So, this means that if we know these operations first and then this operation then we will be able to calculate $A \oplus B$. Now this is the disjunctive sum. Next is bounded sum, bounded sum expression is basically denoted by this where A and B are the two fuzzy sets and the membership values of the resultant fuzzy set it is denoted by this, it is basically obtained using this formula. So, it is basically take the minimum of $\{1, \mu_A(x) + \mu_B(x)\}$ fuzzy set B.

So, this is bounded sum and another is bounded difference. The bounded difference is expressed by this notation and it's membership value can be obtained using this formula. It is the maximum of $\{0, \mu_A(x) + \mu_B(x) - 1\}$. So, we can take the maximum of these two values for each element x, then we can obtain the fuzzy set membership values of the elements which belongs to the bounded difference.

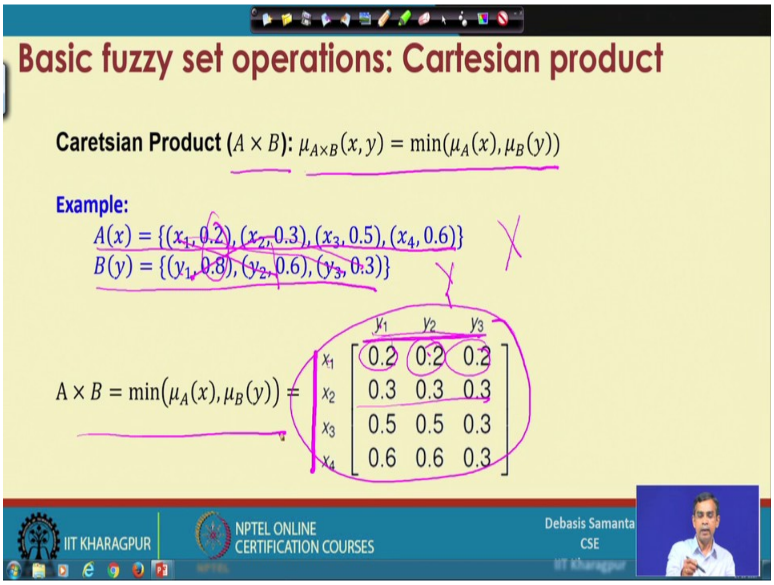
(Refer Slide Time: 14:31)



Now equality and power, these are the other two fuzzy sets. We say two fuzzy sets A and B they are equal and it is denoted by it is denoted by it is denoted by this expression $A = B$. So, two fuzzy sets are equal if for all elements belong to the set has the same membership value, so which is represented by this expression. Now power of a fuzzy set which is denoted as A^α , where α is a constant any value, then the resultant fuzzy sets whose membership value can be obtained using this expression, where this is the membership values of the resultant fuzzy sets and $\mu_{A^\alpha}(x)$ is the original fuzzy sets and here $(\mu_A(x))^\alpha$.

So, it is a exponential operation that can be applied for each elements belongs to the set x then we can obtain the resultant membership values belongs to the set A^α . Now we can note if $\alpha > 1$ then it is called the concentration and if $\alpha < 1$ it is called the dilation. So, these are the two operations that is possible for A and B right.

(Refer Slide Time: 16:05)

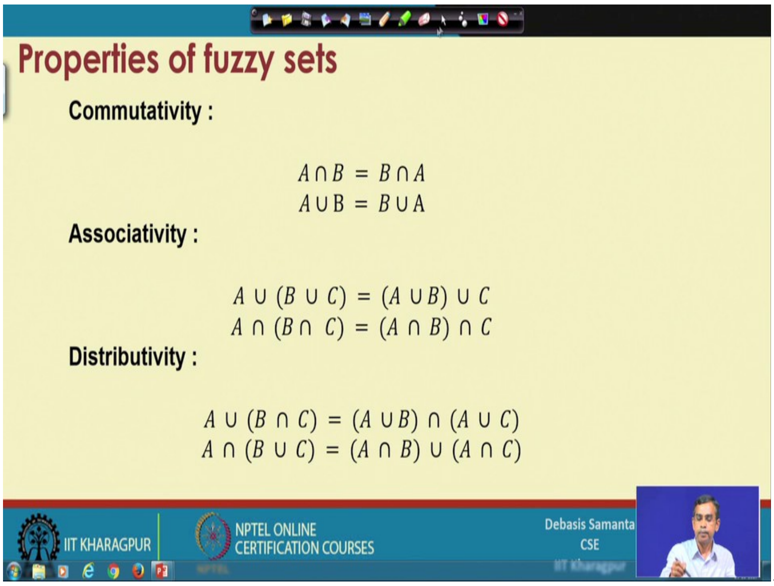


Now, another operation which is very much frequent operation in fuzzy logic it is called the Cartesian product. The Cartesian product of two fuzzy sets A and B is denoted by this notation, denote this notation and it's membership function is denoted by this notation. Now we can note the operation it is basically $min\,(\mu_A(x), \mu_B(y))$. Now here again product whenever we say the product it is basically is a product for all elements. Now for an example suppose x is the A is the fuzzy set, which is defined over a universe of discourse X and B is the another fuzzy set which is defined over another discourse Y.

Then the product $A \times B$ can be defined using, so here x_1 and y_1 and then take the minimum, so it is basically 0.2 and 0.8 so 0.2. So, this can be better represented by means of a matrix. So, all the elements which belongs to the set A can be represented here and all the elements which belongs to the set B can be represented here. And then x_1 and y_1 , so x_1 y_1 and take the minimum in between. x_1 and y_2 , x_1 and y_2 is we are taking the minimum of 0.2 and 0.6. So, it is 0.2. Then x_1 and y_3 , 0.2 minimum of 0.2 and 0.3 and it is 0.3.

Similarly, for x_2 y_1 , x_2 y_2 and x_3 y_3 , so this element will be obtained. So, the Cartesian product can be expressed with the help of this kind of matrix and it is called the relation matrix. Regarding the relation matrix we will learn many things in a later stage. So, this is the operation that can be performed as a Cartesian product. And one thing you can notice the union, intersection or whatever the operation that we have discussed they basically applicable on two fuzzy sets A and B and all the fuzzy sets A and B they are defined over the same universe of discourse. Whereas the Cartesian product can be applied to the fuzzy set they can be over the two or the same universe of discourse.

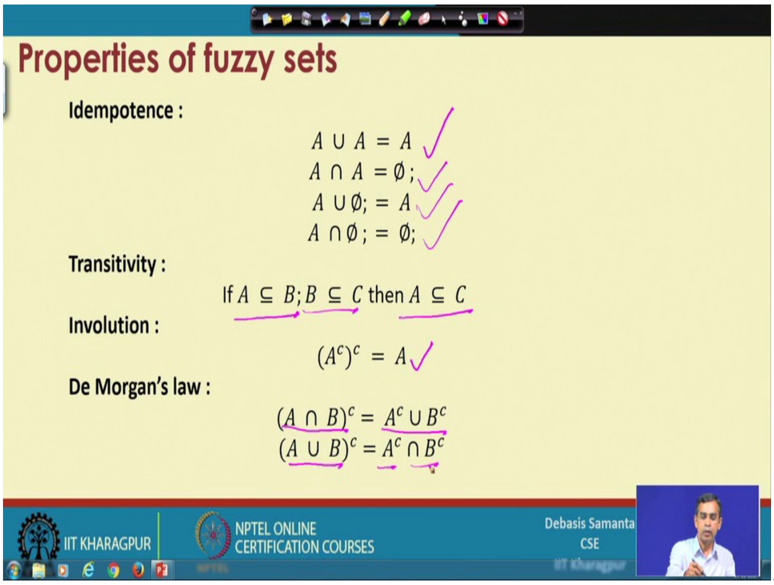
(Refer Slide Time: 18:54)



Now the operations those we have discussed they follow certain properties. The commutative property that it is basically $A \cap B$ is same as $B \cap A$. This is called the commutative product commutative property over the intersection and this is the commutative property over the union operation. Likewise it is called the associativity; that means, this and this are equivalent or this and this are equivalent. The distributivity property also if $A \cup (B \cap C)$ is same as $(A \cup B) \cap (A \cup C)$. So, it is basically A is distributed over B and A is distributed over C likewise for this operation also.

So, whenever the many operations are involved and they can be applied on two fuzzy sets then they satisfy these are the properties it is. There are some other properties also it can be applicable there on the fuzzy sets.

(Refer Slide Time: 20:06)

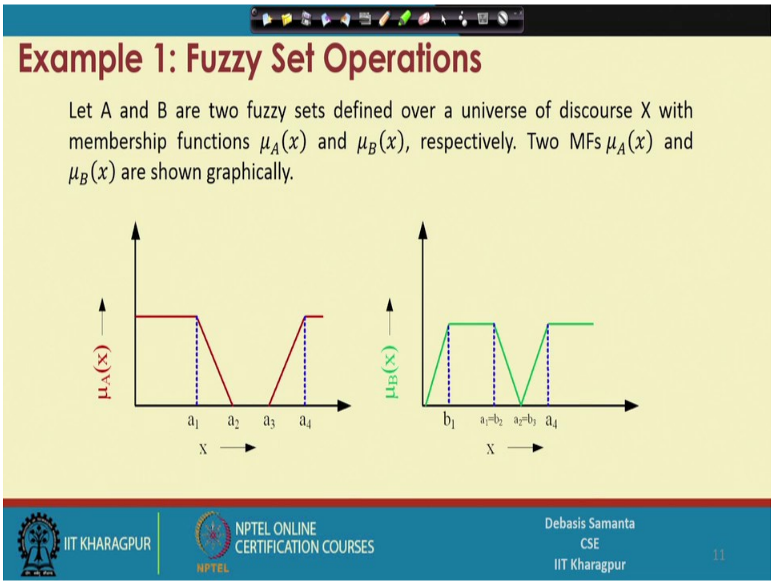


The other property like idempotence; that means, if A is a fuzzy set and if we take the union of the same it will give you the same fuzzy set A. Similarly $A \cap A$ it will give you the null fuzzy set $\emptyset$ and these are the thing simply understandable.

Now, transitivity is basically if $A \subseteq B; B \subseteq C$ then we can say that $A \subseteq C$. So, this is the similar operation that is applicable for the crisp set. Now involution, these operation be is if we take the complement of the complement of the same fuzzy set then it will return the original fuzzy set. The De Morgan's law we know which is there in Boolean logic as well as the crisp logic. So, it is basically $A \cap B$ and if we take the complement it is same as $A^c \cup B^c$, alternatively if $A \cup B$ take the complement it is equivalent to $A^c \cap B^c$.

So, these are the properties those are hold good for fuzzy sets. These properties are very much useful whenever we want to perform many operations on the different fuzzy sets.

(Refer Slide Time: 21:34)



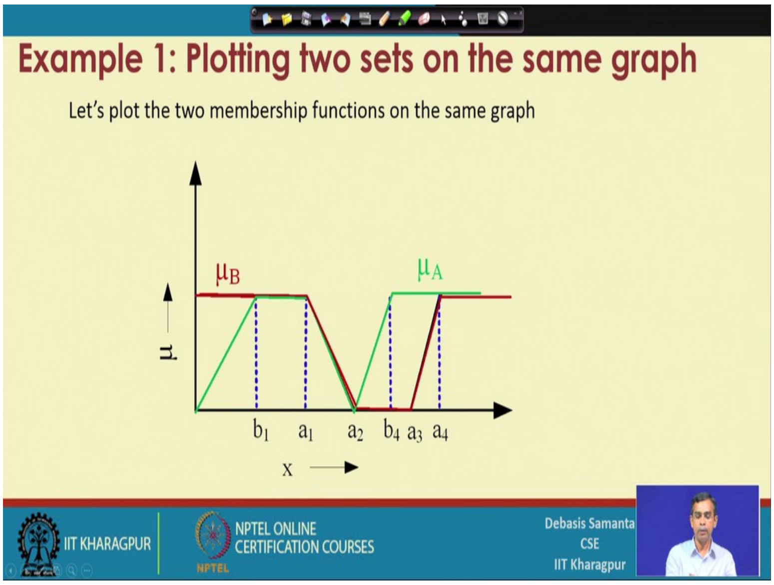
And one thing we can note, the operation that we have discussed about it basically if two fuzzy sets are given then how we can obtain another fuzzy set. So, if A and B is known to you then we can obtain the another fuzzy set C, now if we know C then we can help we can use another operation to find another. So, those operations are basically produce many other fuzzy sets from the given fuzzy sets.

Now I would like to elaborate the fuzzy set operation more clearly with some figures, is a basically called the graphical representation. So, sometime in order to understand the fuzzy operation graphical representation of the two things is more useful and there are many tools which basically follow the graphical way of representing fuzzy set and then performing their resultant operation.

Now, here first example here this is the membership function of. So, this this figures shows the fuzzy set A whose membership function is shown by this graph. And another fuzzy set B whose membership function is shown using this graph. Now, we want to find many operation like say union like say intersection whatever it is there. Now in order to find the union or intersection or complement of course, complement is not required here only say union or intersection let us suppose

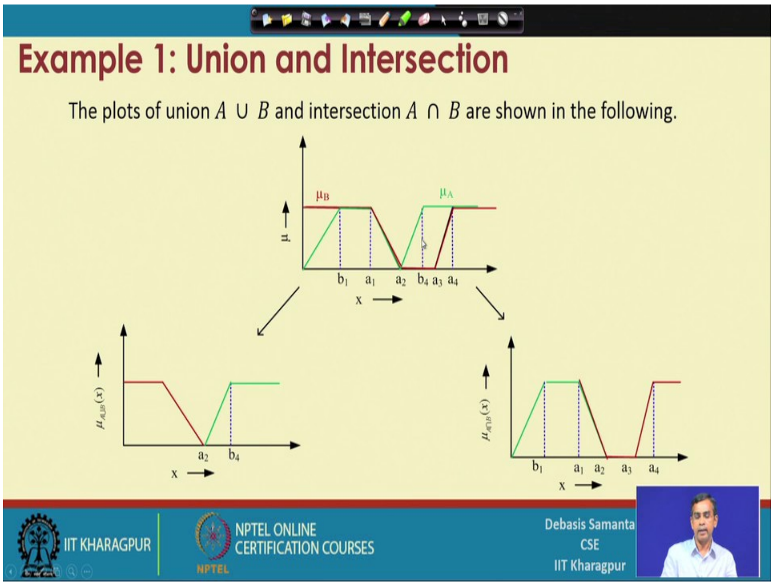
Then best idea what is we can draw the two graphs on the same graph. Now for example, if we draw the graph of the two graphs

(Refer Slide Time: 23:20)



the resultant graph will be obtained like this. Now once the resultant figure is available then we can easily identify if the union or it is the intersection of the fuzzy sets.

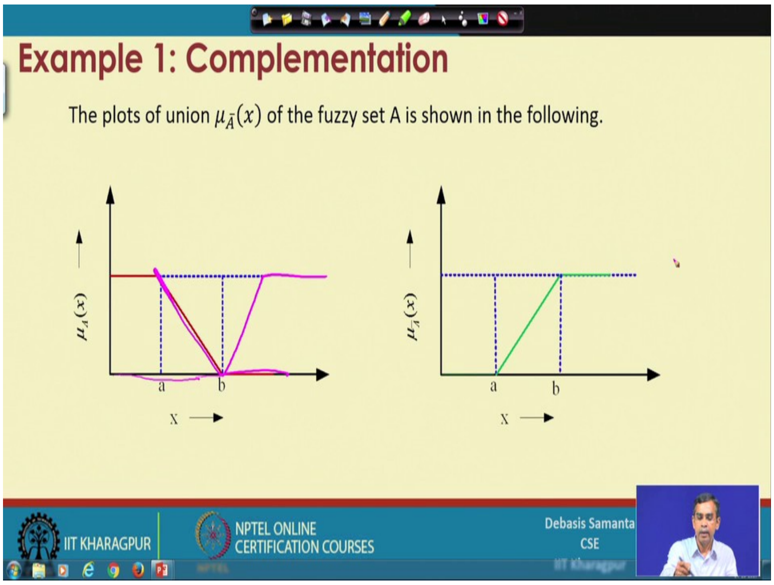
(Refer Slide Time: 23:35)



So, here if this is the graphs of the two fuzzy sets on the same plot, then we can have the union union operation, it is basically union operation union operation we take the max of this one. So, upto this part this is the max part and then this for the next part. So, this why this is the union resultant this one.

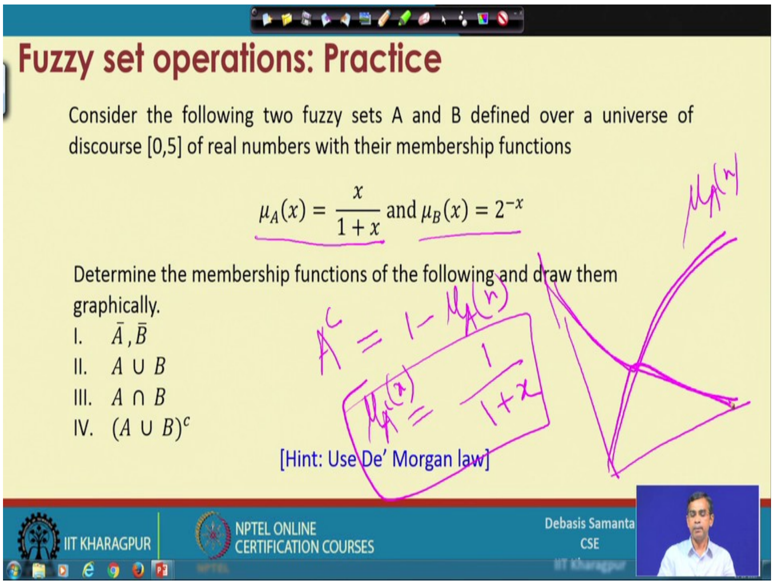
Similarly, for the intersection we take the min this and then this. So, basically this is the graph for the intersection. So, what I want to say that, if we can plot the membership function of two graphical representations, then from there we can obtain the resultant fuzzy sets that is also graphically.

(Refer Slide Time: 24:27).



Now, here again another example so this is the fuzzy sets A and it's complement is basically complement will be this one. This one is the complement. So, it is basically shown it is there. So, graphical way of representing their operation is sometime more useful to understand. So, we have discussed it.

(Refer Slide Time: 24:54)

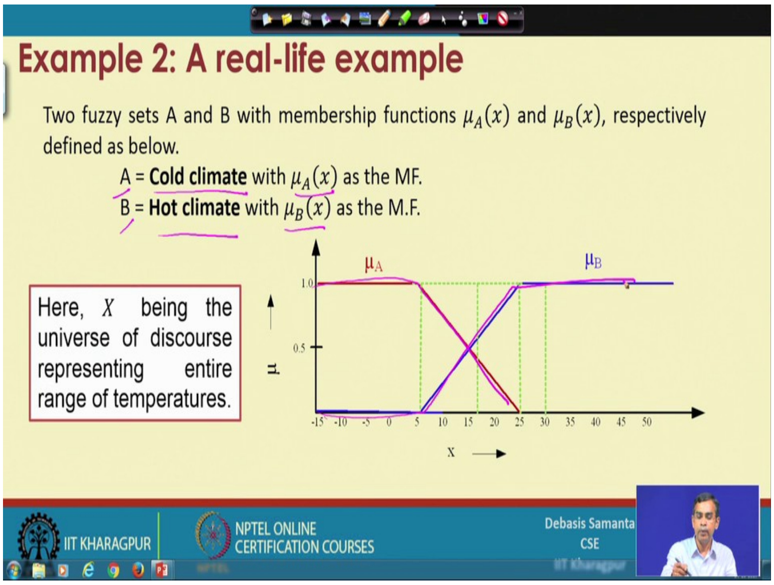


Now we will discuss about few example. So, that we can clear about our idea. Sometimes with a graphical representation we can express the operation, sometime also mathematically you can express the operation.

For example, say suppose A is a fuzzy set A is a fuzzy set whose membership function is defined by this expression, we can plot a graph so graphical representation can be obtained accordingly. Similarly another fuzzy set B which is shown here. Now if I if we want to know A complements. So, A^c this one this is equals to $1-\mu_A(x)$. So, this means is basically $\frac{1}{1+x}$. So, this basically we can say this is $\mu_{A^c}(x)$ this is the membership function of the resultant fuzzy set.

Now, likewise $\mu_B(x)$ and it is complement can be obtained and graphically if we plot then $\mu_A(x)$ can be plotted like this. So, it is $\mu_A(x)$ and $\mu_B(x)$ that can be obtained like this the $\mu_B(x)$. Then the union of the two fuzzy sets A and B is basically this is the union and if the intersection then intersection can be obtained by this one. So, graphically both can be obtained as well as mathematically it can be obtained.

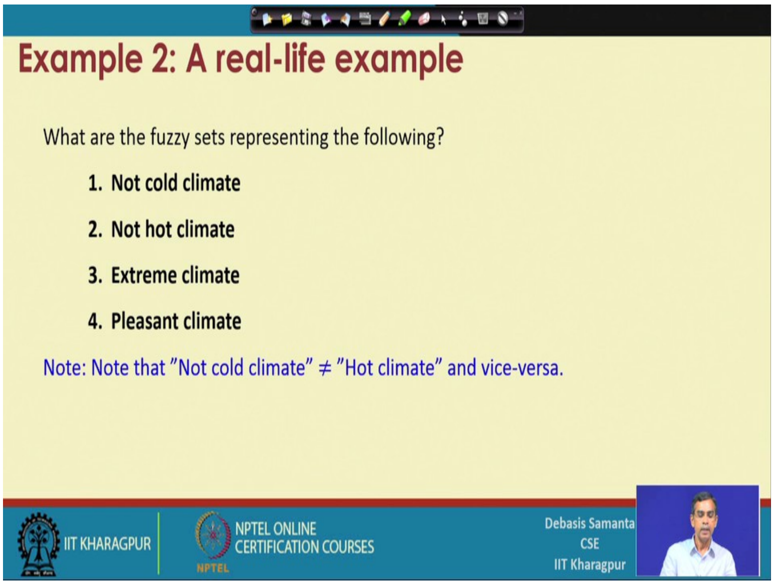
(Refer Slide Time: 26:40)



Now, the idea about this fuzzy operations is more meaningful. Now, we know every fuzzy set is basically expressed by certain meaning, this is called the linguistic hedge. Now for example, say suppose A is a fuzzy set and B is another fuzzy set. A basically the cold climate and B is the hot climate representing, with a membership function $\mu_A(x)$ and $\mu_B(x)$. So, that two fuzzy sets can be representing the graphically using this set. This is the A and this is the set B.

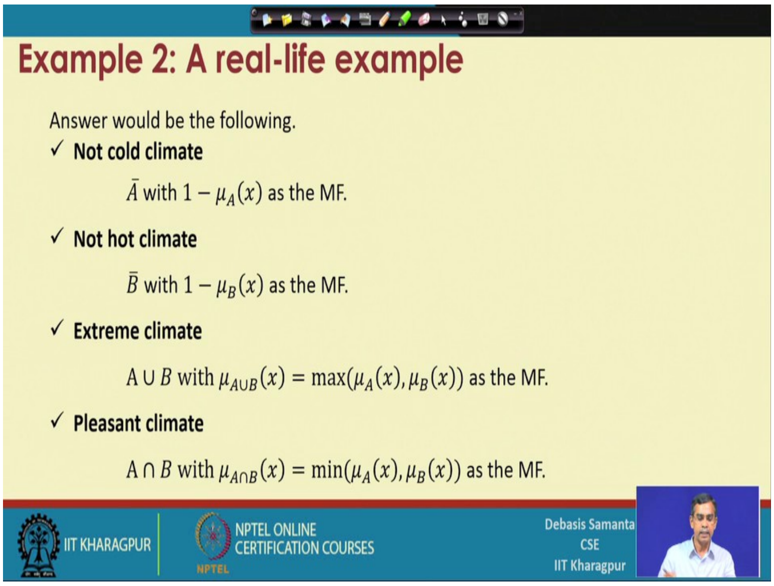
Now, there are. So, operations whenever we perform on A and B has the meaningful representation that can be obtained like this.

(Refer Slide Time: 27:26)



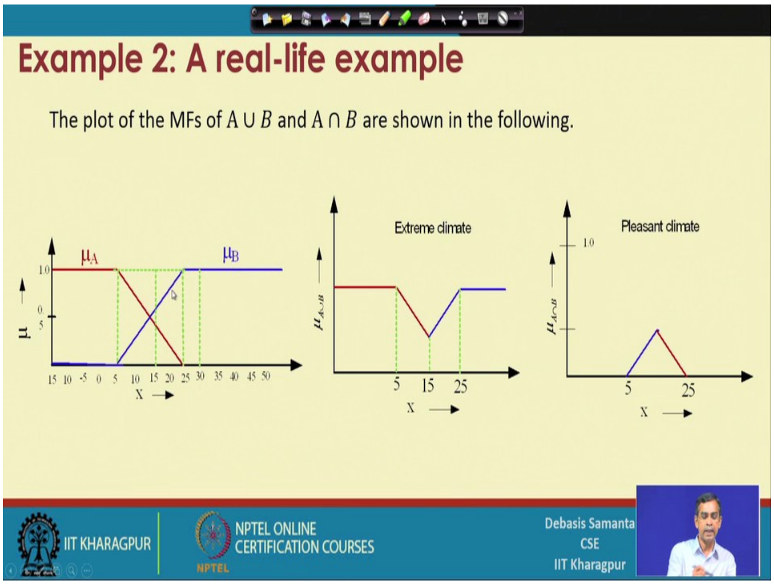
Say, if we know the fuzzy set cold climate or hot climate, then we can know the fuzzy sets not cold climate. It is basically complement of the cold climate or not hot climate is basically complement operation of the hot climate. Extreme climate is basically is the operation of both; that means, it is union and pleasant climate is basically intersection.

(Refer Slide Time: 27:52)



So, this is represented here by this. So, not cold climate is this one, not hot climate is a complement B and extreme climate is union and then pleasant climate is this one. Now graphically the same thing can be shown also here.

(Refer Slide Time: 28:12)



If so these are the two fuzzy sets A B. The extreme climate that can be obtained by the plot, this is the resultant graph of the fuzzy set extreme climate and this is the resultant fuzzy set of the pleasant climate.

Now, this example basically shows you that how the different operation is meaningful in the context of fuzzy sets. Now let us stop it here. So, these are the different fuzzy set operation and we will discuss the other things in the next lecture.

Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 05
Fuzzy relations

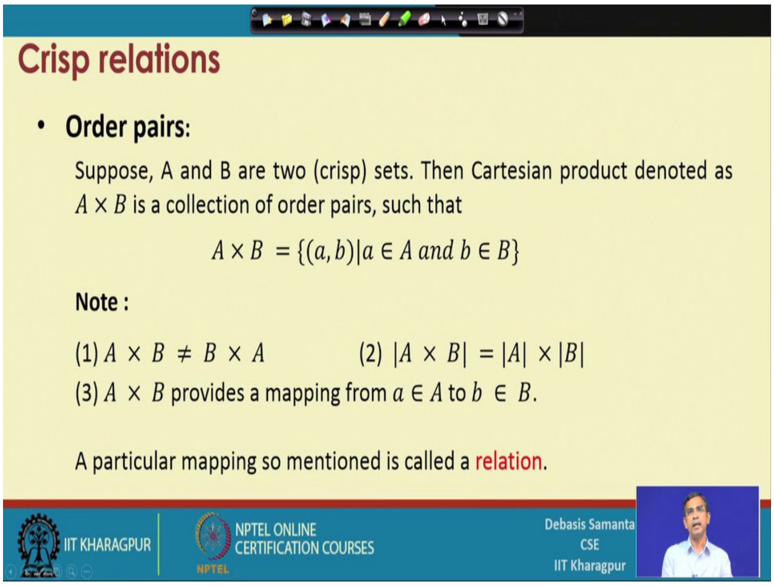
So, in the last lecture we have learned about different operations on fuzzy set. So, different operation on fuzzy set that we have learnt is basically given a two fuzzy set or a fuzzy set how another fuzzy set can be obtained. Now we are going to learn another concept in fuzzy logic it is called the fuzzy relation. So, by means of fuzzy relation we say that if one element belongs to a fuzzy set then how this element is related to another fuzzy set. So, this basically the relation; between the two elements, which belongs to the two different fuzzy set.

(Refer Slide Time: 01:05)

So, fuzzy relations in many way related to the crisp relation. Crisp relations means the relation those are there on the crisp set. So, will first learn about the crisp relation and then whatever the operations those are possible on crisp relation is basically semi applicable to the fuzzy relations also, but there are certain difference. So, it will be better if will learn first operations on crisp relation and then the operation that can be applied to the fuzzy relation. Some examples also will be considered in order to understand the crisp relation and then we will be in a position to know fuzzy relation. Some examples operations those are possible on fuzzy relation and finally, we clear our idea with some examples.

So, these are the topics that we are going to cover in this lecture.

(Refer Slide Time: 02:02)



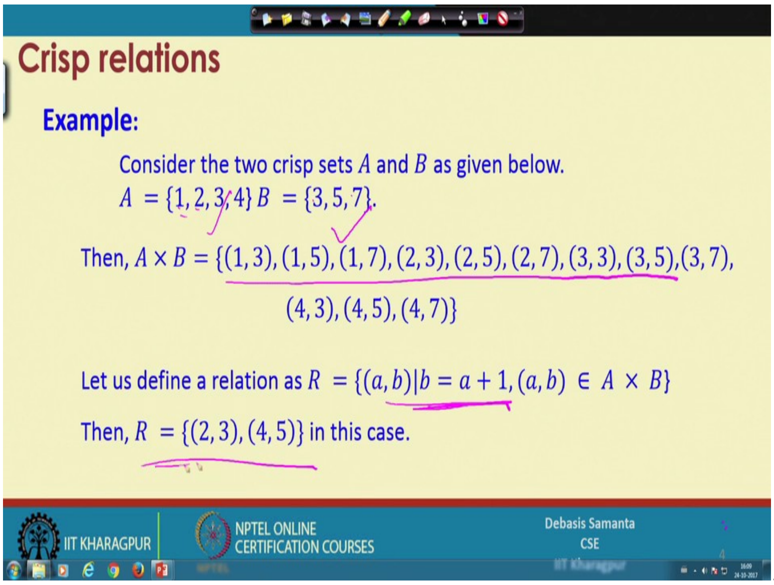
So, let us first discuss about the crisp relation. Now crisp relation basically is an collection of order pair. So, if A and B are the two sets then it's order pair is denoted by the Cartesian product $A \times B$ and it basically gives the collection of order pair $\{(a,b) \vee a \in A \wedge b \in B\}$.

So, this order pair relation is important to understand the crisp relation. Now a particular mapping is basically belongs to a particular relation, and we know so far the crisp relation is concerned these are the different property holds good. The first property is that $A \times B \neq B \times A$; that means, they are not commutative and here is basically the number of elements which is belong to the product is same as the number of elements belongs to the constituent set A and the product of the number of elements belongs to the set B.

So, this also the equation hold goods and as I told you $A \times B$ essentially provides a mapping the from a set from an element $\in A$ to another element $b \in B$. So, it is basically a mapping and this mapping is expressed by means of an order pair and this particular mapping is called a relation.

Now, we can understand this relation better if we consider an example.

(Refer Slide Time: 03:50)



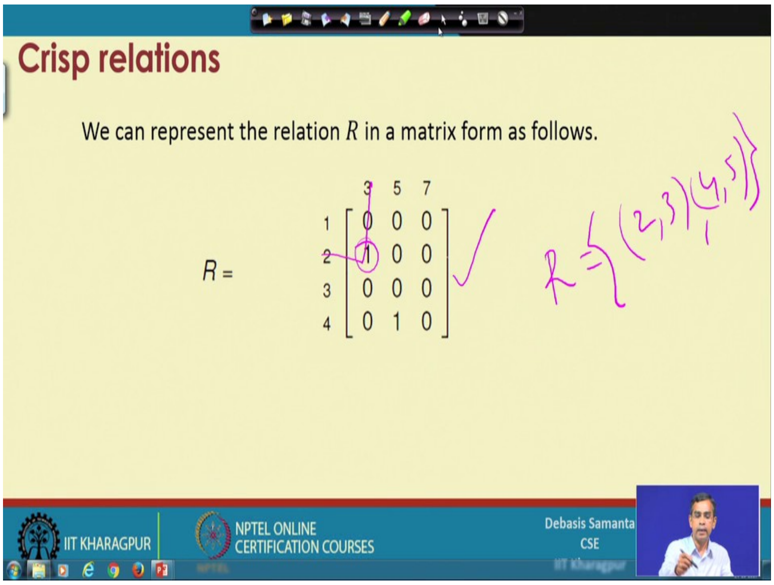
So, example suppose two crisp set A and B and A is represented by this form and B is {3, 5, 7}. So, these are the two sets A and B and we can obtain their Cartesian product. Cartesian product we can obtain for this two phrase is all possible order pairs. So, it is shown here. This is the Cartesian product of that two fuzzy set A and B. So, here for given {1, 2, 3} then $(1,5),(1,7)$ then $(2,3),(2,5),(2,7),(3,3),(3,5)$, $(3,7)$ and so on.

So, these are the different what is called the elements which belongs to the Cartesian product of the two fuzzy sets A and A and B and then the relation, I told you the relation is basically a particular mapping. Now here we express the relation and this is a this is this is suppose the relation that we have discussed here.

So, relation between the 2 order elements in a order pair should satisfy the this equation, if it is satisfy then it gives a particular set or it is basically a relation which is shown here. For example, if this is a relation hold good for every element then the relation that can be obtained is basically this one.

So, a relation is basically a collection of order pairs which satisfy a particular mapping or a particular definition.

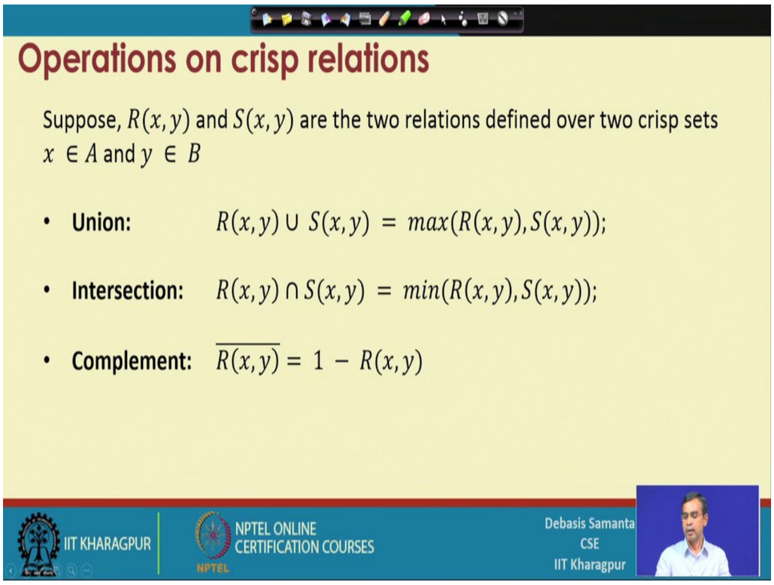
(Refer Slide Time: 05:39)



Now, so this is the crisp relation and such a relation can be expressed in a more compact way and this compact way it is called the matrix representation of a relation. Now the matrix representation of the relation which we have learned earlier, so R is (2, 3) and (4, 5) is denoted by this matrix. You can see 1 and 3 which is not belong to the set, so it is 0 and the elements 2 and 3 belongs to the set, so it is 1. So, here 0 and 1 are the entries in the relation matrix. 0 indicates that that order pair is not belongs to this relation and 1 indicates that that order pair belongs to the relation.

So, this way a relation can be represented by means of a relation matrix.

(Refer Slide Time: 06:44)

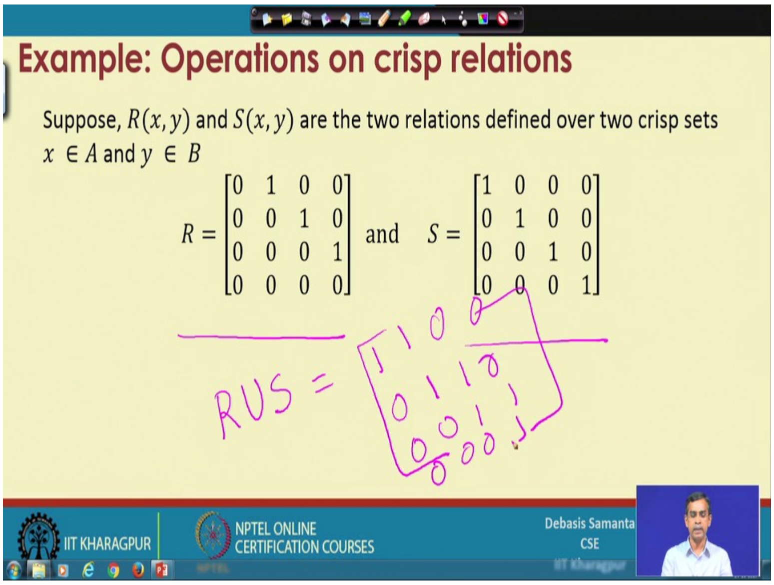


Now, so operation on crisp relation so these are the relations if it is available to us we can apply different operations on it. Say suppose R and S are the two operations defined over x and y, where x is the some some elements belongs to the universe of discourse x and y is the element belong to the universe of discourse y. So, $R(x, y)$ can be obtained as a relation matrix similarly $S(x, y)$ can be obtained by means of an another relation matrix.

So, if the two relation matrix, are available to us then we can apply many operations on them. So, the operation on them that can be a applied can be applicable is union, intersection, complement like this. Now you can find the difference between the union of two fuzzy sets and union of two fuzzy relations. So, union operation of two fuzzy set is expressed by this form. So, R is the relation and B is the S is the another relation right and this relation obtained over the two crisp set say A and B then the union of the two relations can be defined as a max of the two entries in x and y in both relation.

So, one example can be given here. So, this is the union operation likewise the intersection operation is basically minimum values of the entries and the complement is 1 minus the entries in each elements.

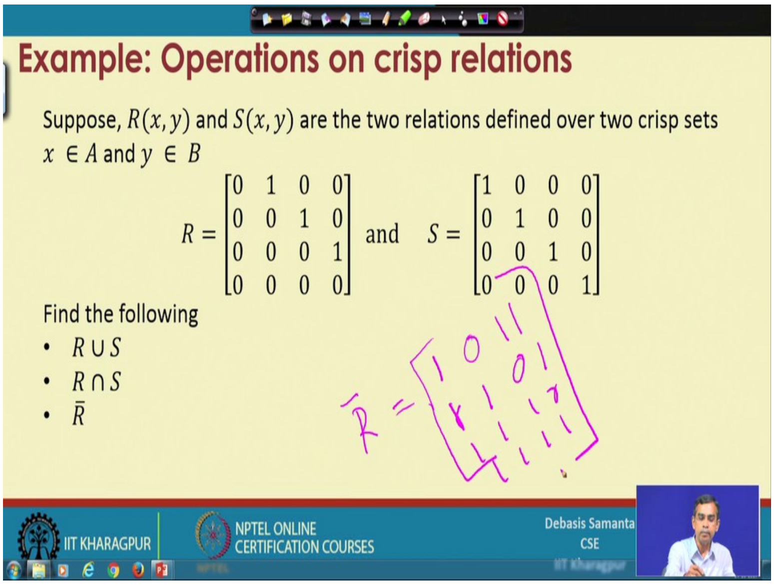
(Refer Slide Time: 08:41)



Now, one example that can be consider here. Suppose this is the one relation that can be obtained over A and B and this is the another relation obtained over the sets A and B. So, it is basically $A \times B$ with some relation it is also $A \times B$ with different relation and then we want to obtain union of the two. So, union of the two we can write $R \cup S$. So, it is basically one matrix now union operation as I told you it is a max. So 0 and 1, so you have to take the 1. 1 0 it is a 1 then 0 0 for the first rows. Similarly 0 1 1 0, 0 0 1 1 and 0 0 0 1.

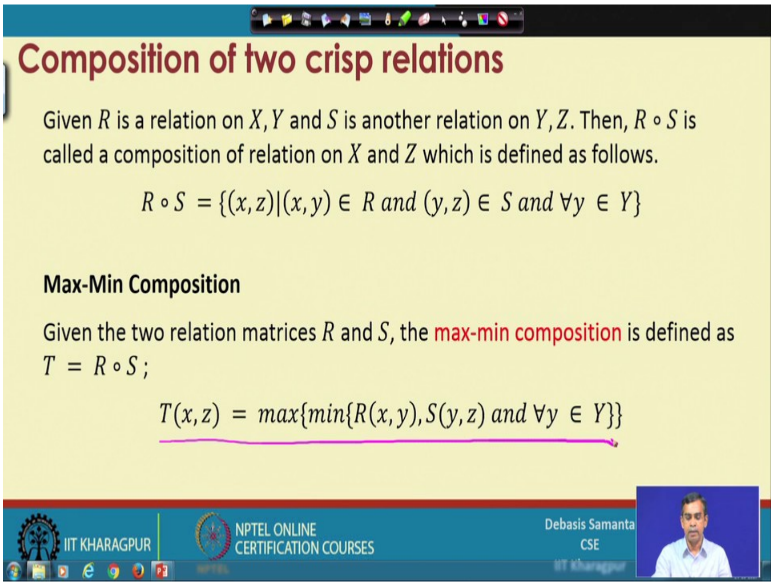
So, this is the another relation matrix that can be obtained using the operation relation operation of the two relations R and S. So, this way we can obtain the relation.

(Refer Slide Time: 09:52)



Now, using the same concept you can find easily the union of two, intersections of two and then complement. Now, so far the complement operation is concerned $\acute{R}$ this is equals to it is basically complement value. So, if the 0 then it will be 1 and it is 1 then 0. So you can complement of these is this this this and then 1 1 0 1, 1 1 1 0 and 1 1 1 1. So, this is the complement of the relation R.

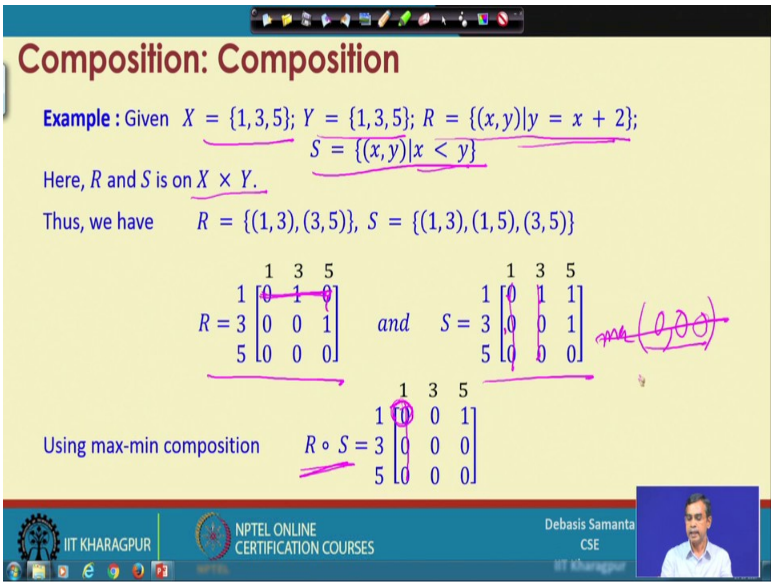
(Refer Slide Time: 10:45)



Now, there is another important relation it is called the composition and composition relation is why they applicable in the context of fuzzy relation. So, composition operation it is denoted by this symbol $R \circ S$; that means, from two relation we can find another relation, but. So, this R relation suppose over two set, A and B and S relation is over another say, A and C. So, here C is the one common set then we can obtain $R \circ S$ basically relation from A to B via C. So, this is the concept that is called the composition.

Now, composition operation can be defined mathematically using max-min calculation. The max-min it is called the max-min composition this is why and is denoted by this expression. Max-min composition is basically this one. Now it is little bit difficult to understand at the moment. So, I can give an example. So, that you can understand it basically follows the similar concept of product of two matrix actually. So, it basically take the first minimum of corresponding entries and then take for a particular entries the maximum value.

(Refer Slide Time: 12:08)

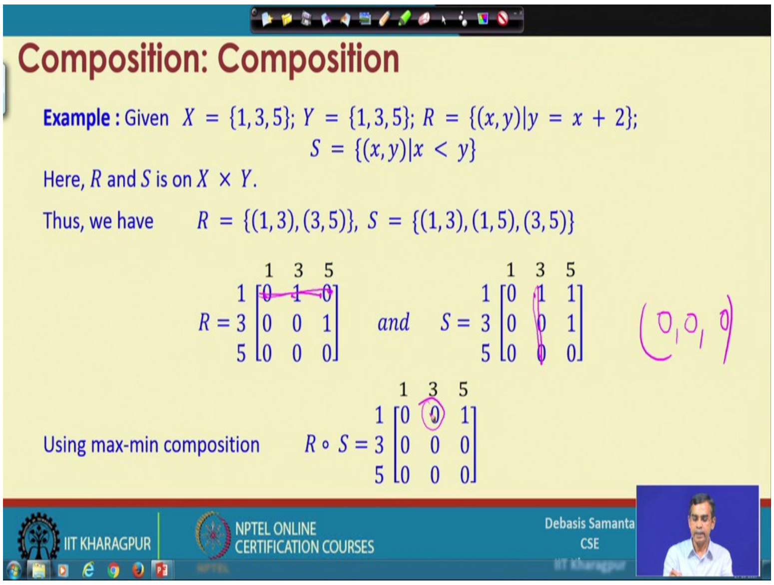


So, let us have some example. So, that we can understand the max-min composition or simply a composition operation to relation. Now this is a one example we have to consider carefully. So, suppose x is the one universe of discourse y is another and the relation $R(x,y)$ defined over x and y and S is also another relation defined over the same discourse x and y. The relation that is there for a R it is basically this is the relation and this is the relation that is there in S.

Now, based on this thing we can easily obtained the Cartesian product and then applying this relation we can obtain this matrix and this matrix. So, these are the two relations obtained from the two fuzzy set through crisp set x and y. Now having this relation we can find the composition of the two. Now composition basically we take first row wise and then column wise just like a product to obtain the first element here. Sorry, so this is the row and this is the one to obtain the first element here.

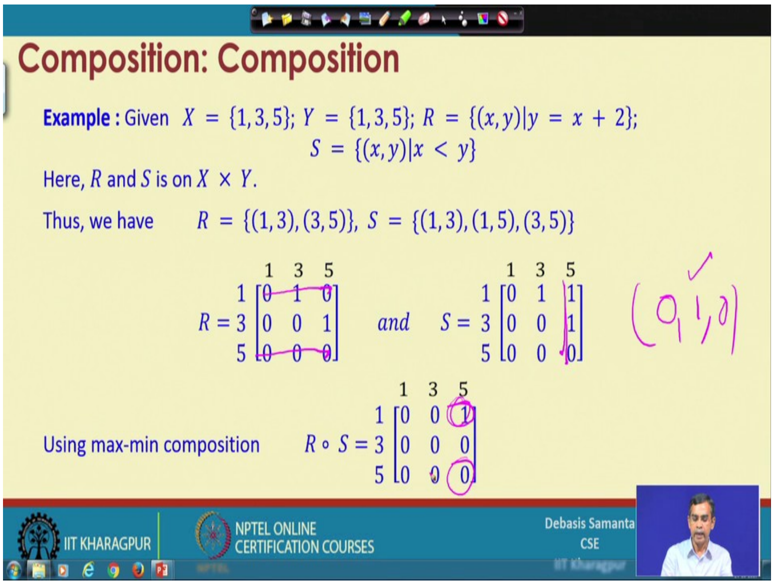
So, is basically take like, so 0 and 0 take the minimum, so minimum is 0. So, 0 is a minimum. Then 1 and 1 as 1 and 0 then it is a minimum is 0. Then 0 and 0 so minimum is 0 and then take the max. Max of this so this is 0, so this is a 0. For the next element we can obtain so this and then this one. So, 0 and 1 so further next further next this one and then this one, so you can get this element. So, this and this we can go so; that means, 0 and 1 take the minimum it is 0.

(Refer Slide Time: 13:58)



And then 1 and 0 take the minimum it is 0 and then 0 and 0 take the minimum this one and then maximum. So, it is 0.

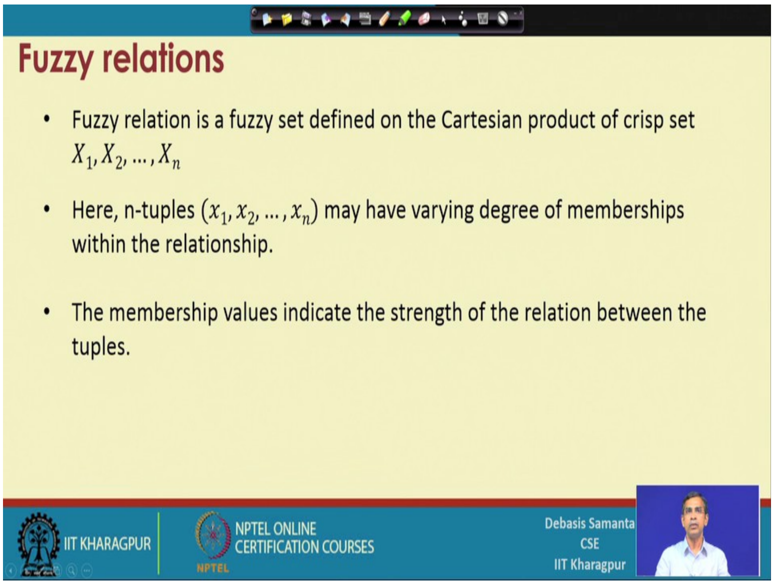
(Refer Slide Time: 14:17)



Now, let us see how this element can be obtained. So, again we have to take this one and then this one. So, first 0 and 1 0 then 1 and 1 1 and 0 and 0 it is 0 and taking the maximum this one. So, it get this one. So, these element, again we can be obtained if we take these one and the this last element can be obtained. So, this way we can obtain the relation called the max-min composition of the two relations $R \circ S$.

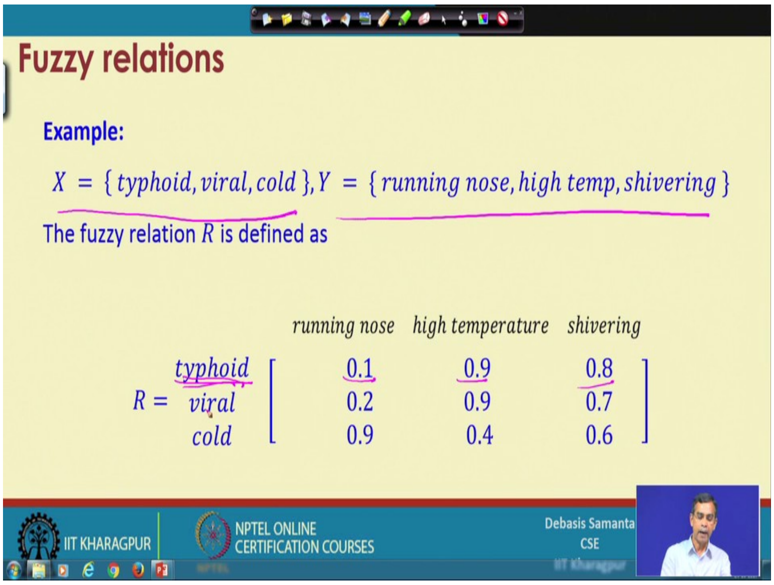
So, it needs a little bit practice to understand it. So, it will basically take the min corresponding to this traversing and then taking the max of all these will give a particular element. So, this is the idea and now let us see, how these operations those are applicable to the crisp is also applicable to the fuzzy, but in a different one.

(Refer Slide Time: 15:10)



So, difference between the fuzzy relation and crisp relation lies in the terms of increase in the relation matrix. In case of crisp relation the entries in the matrix is either 0 or 1 whereas, in case of fuzzy relation the entries in the matrix is any value in between 0 and 1 both inclusive.

(Refer Slide Time: 15:24)



Now, let us start with an example about the fuzzy relation. Suppose two fuzzy sets which is described over the 3 elements which is here. One fuzzy set is X and another fuzzy set is Y. The different elements in the fuzzy sets are here. In X it is {typhoid, viral, cold}. In fuzzy set

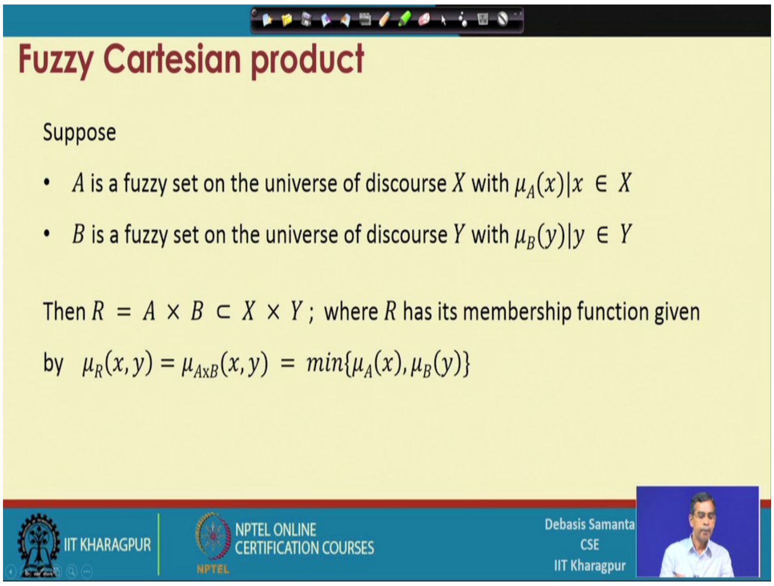
Y, the elements are they are {running nose, high temperature and shivering}. Now you can understand what is the meaning of these two fuzzy sets and the relation then.

So, basically here if the disease is typhoid then what are the difference symptoms are there. So, if the disease is typhoid the symptom that running nose is, but with the strength 0.1, high temperature 0.9 and shivering 0.8.

So, every disease and the different symptoms with the different membership values is represented and by means of a relation matrix. So, if it is a viral then what about the shivering. The shivering is 0.7, if it is a viral then running nose, but running nose with little bit less uncertainty than the shivering that is 0.2 and 0.7. So, the relation matrix basically shows the different element which are belongs to the different sets how they are related to each other. So, this basically the physical significant of the fuzzy relations and now one thing it is clear that the element in the fuzzy relations; that means, the entries in the relation matrix is basically any value in between 0 and 1 both inclusive.

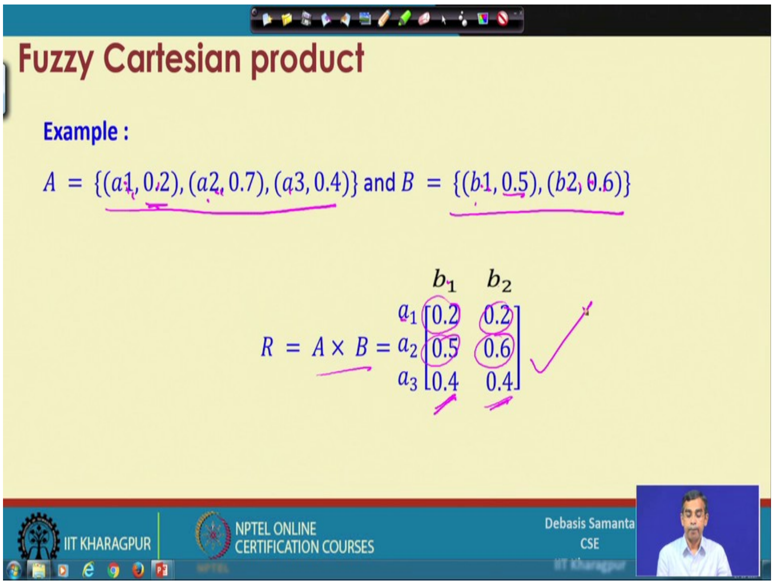
So, this is the only difference between the fuzzy relation and the crisp relation, otherwise every operation those we have defined in case of crisp also equally applicable to the fuzzy sets. Now let us see what are the different operations they are possible for the fuzzy relations.

(Refer Slide Time: 17:45)



So, here so the fuzzy relation again defined as in the crisp relation like the min operation. So, it is the min actually. So, this is the min operation.

(Refer Slide Time: 18:00)



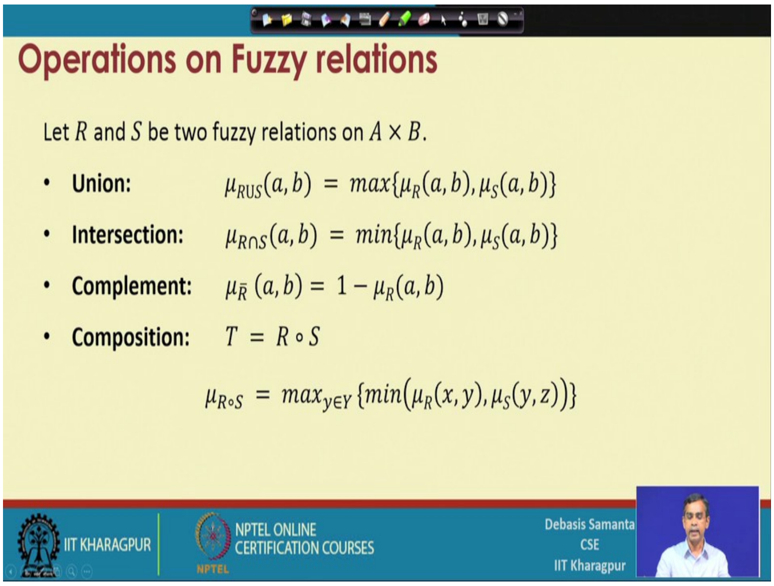
Let us see one example. So, here say A and B are the two fuzzy set. A and B are the two fuzzy sets. Now I can find a relation. The relation can be obtained as we have discussed here, now relation operation that can be defined over two fuzzy sets it is basically represented by this expression that $\mu_R(x, y)$ where is a membership values belong to relation and it is denoted as $A \times B$ as I told you Cartesian product of $x\,y$ and it basically takes the minimum of two corresponding values in both the set A and B for x and y respectively. So, it is basically taking the minimum.

Let us have an example, so we can clear our idea. This is an example can be followed to explain the relation operation for the fuzzy set in terms of Cartesian product A. Now here A is the set which is defined like this. B is another set which is defined like this and then the relation are is basically Cartesian product as I told you. So, it basically for a_1 and b_1, a_1 and b_2 so $a_1 b_1, a_1 b_2$ now for $a_1 b_1$ we have to take the minimum. So, 0.2 and 0.5 take the minimum, so it is a minimum entries. Similarly a_1 and b_2 0.2 and 0.6 take the minimum so 0.2.

Likewise a_2 b_1 so minimum is 0.5. a_2 b_2 the minimum is 0.6 a_2 . So, this is a_2 now a_3 and b_1 so 0.4 and $a_3 b_2$ 0.4. So, these way we can obtain the relation matrix taking the min operation that is there. So, this way we can obtain the relation if the two fuzzy sets are given to us.

Now, let us define different operations on fuzzy relation like the different operation in crisp relation. So, like union, intersection and then complement these are the operation.

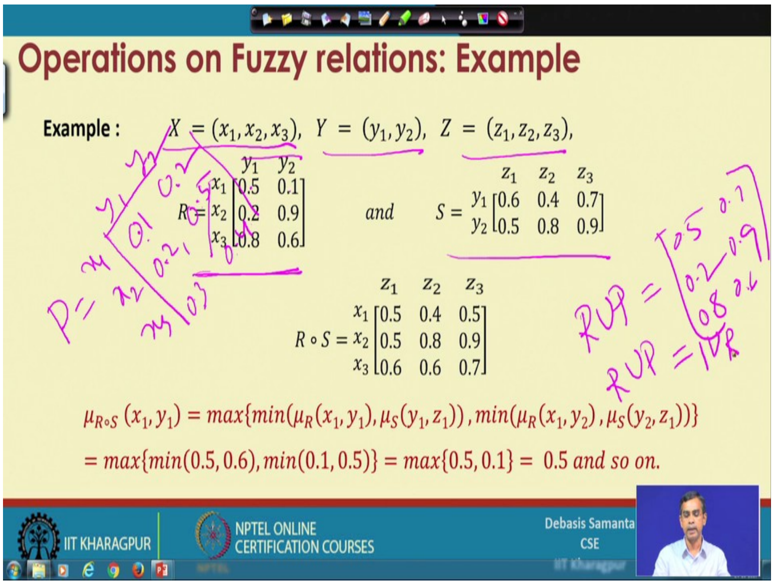
(Refer Slide Time: 20:21)



So, union operation on two fuzzy sets can be defined using this expression. It is basically taking the maximum value of the two entries there. So, it will give the new matrix taking the maximum of the corresponding entries. Intersection basically taking the minimum of the two entries and complement it is a unary for one operation. So, you take the if it is $\mu_{\dot{R}}(a,b)$ then the value or entries in the relation matrix will be the complement; that means, $1-\mu_R(a,b)$.

Now, some example can be followed to understand this concept. Another composition I will discuss the composition operation in details in regards the fuzzy sets it is basically same as max-min composition.

(Refer Slide Time: 21:18)



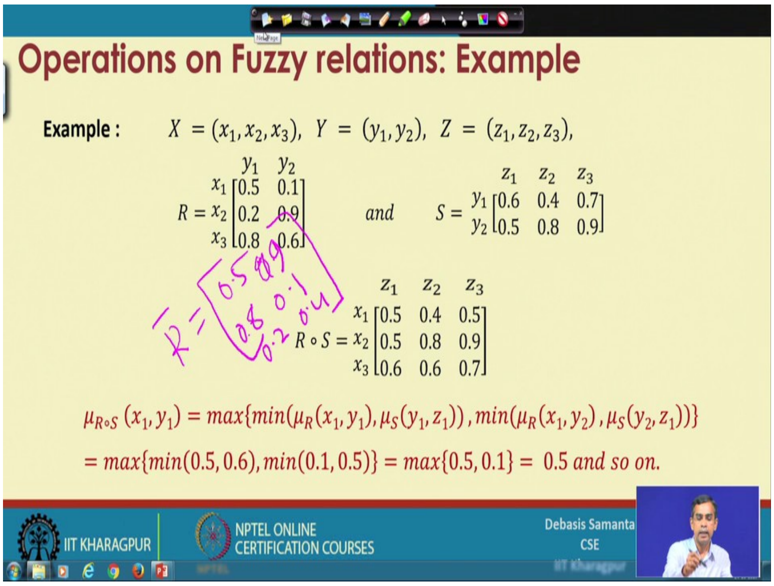
Taking the similar concept now better we if we have one example. So, X is a crisp set Y is another and Z is another. So, we can consider they are the universe of discourse of for the fuzzy sets may be and so R this is the one relation defined over two sets which is discussed over the universe of discourse X and Y. So, this is the relation over the two fuzzy sets and S is the another relation which is defined over Y and Z.

So, these entries are given to you. Now if it is given to you then we can calculate $R \cup S$ easily. Now again here we can say $R \cup S$ here basically elements those are not same that is why we cannot apply $R \cup S$, but the two I mean union operation is applicable if the two relations are defined over the same elements. So, if this relations is defined over this one this one then another relation should be defined this one then only we can. For example, suppose I defined another relation P and $x_1, x_2 \wedge x_3$ and $y_1 \wedge y_2$ and say 0.1 0.2 0.2 and 0.5 0.3 and 0.4. So, this is a relation.

Now, if we want to find the union of the two relation. So, $R \cup P$ then relation basically taking the maxima of the corresponding entry. So 0.1 and 0.5 so in the first entry 0.5 and then 0.2 and 0.1 so you should take 0.2, 0.2 and 0.2 so 0.2 and 0.5 and 0.9 0.9. Then 0.3 and 0.8 0.8 then 0.4 and 0.6 it is 0.6.

So, this is basically the relation obtained over the union operation of the two relation R and P. And you can note that $R \cup P$ this is equals to same as $P \cup R$, it means they holds the commutative property. Now likewise the intersection. Intersection we have to take in, so it is basically take the minimum of this on union where is the maximum and intersection is the minimum.

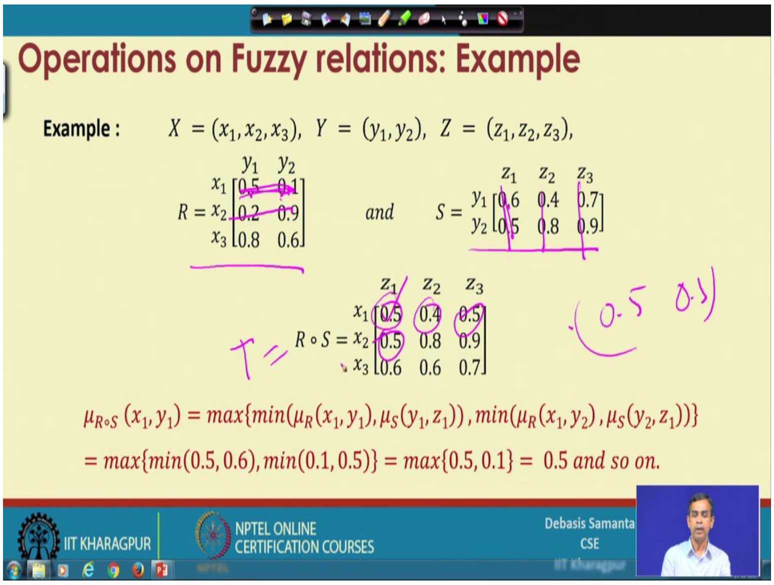
(Refer Slide Time: 24:18)



And then complement also can be obtained. For example, so complement operation this $\acute{R}$ that can be obtained like. So, 0.5 then 0.9 then 0.8 then 0.1 0.2 and 0.4. I hope you have understood how it is obtain it is basically taking the complement $1-0.5$, $1-0.1$ and this way.

So, this way the complement operation over a relation R can be obtained. Now this is another example that I am going to discuss is a composition. So, R and S are the two relation.

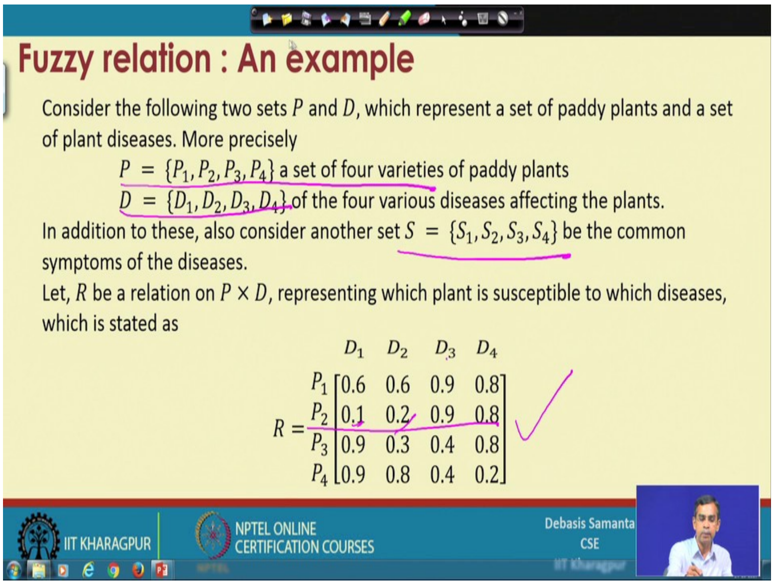
(Refer Slide Time: 24:56)



And we want to find another relation T which is a composition of $R \circ S$. So, using the max-min composition it is the composition that we have discussed about the Cartesian product finding. So, take the minimum first and then minimum of all take the maximum again we can follow it is these one and these traverse. So, it will give this one; that means 0.5 and 0.6 we have to take the minimum so it is 0.5 and 0.1 and 0.5 we have to take the minimum 0.1 and taking the maximum of this so 0.5. So this way 0.5 and likewise if we traverse this one and this one the these element can be obtained. these one and these one then this can be obtained these then this element can be obtained and so on.

So, this is the max min composition operation that can be applied on two relation R and B.

(Refer Slide Time: 26:02)

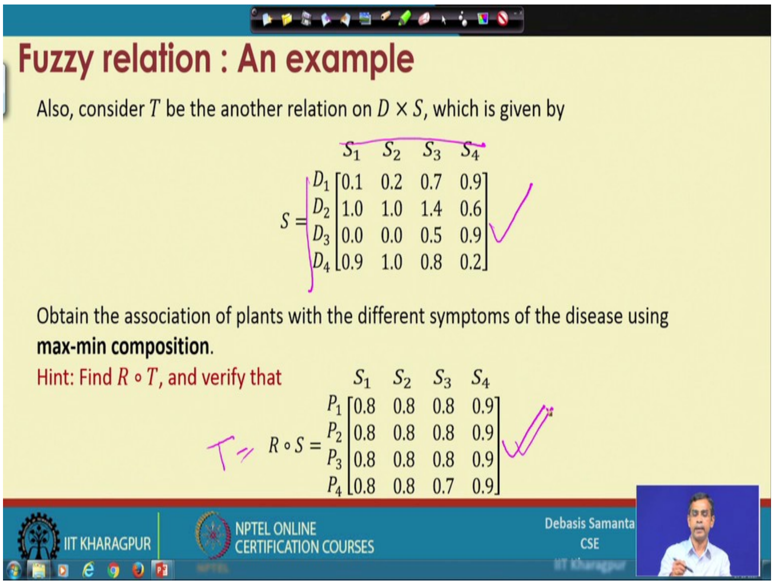


So, this is the relation operation and I can give another last example in this direction. Here these examples is very interesting to note. So, P is the set with different element $\{P_1, P_2, P_3, P_4\}$. D is the another set with different element say $\{D_1, D_2, D_3, D_4\}$. Now in the context of our real application, so P basically consider a set of varieties of paddy plant; that means, P_1 is a one type of paddy plant, P_2 is another and so on.

Now, D the set D represent the different diseases where D_1 is a type of disease, D_2 is another and so on and say S is the another set with basically set of symptoms, the symptoms are $\{S_1, S_2, S_3, S_4\}$. Now how a particular plant is related to the disease that can be given by means of a relation. So, this is basically a fuzzy relation showing that how a particular plant is related to the different disease that may have. For example, P_2 is a plant and is susceptible to disease D_1 with the 0.1 certainty, D_2 is 0.2, D_3 0.9 and D_4.

So, we can say that P_2 is very much susceptible to the disease D_3 or D_4 and less susceptible disease D_1 D_2. It is the concept. So, these are meaning of the fuzzy relation. Now having this fuzzy relation are we can obtain another relation by means of composition operation.

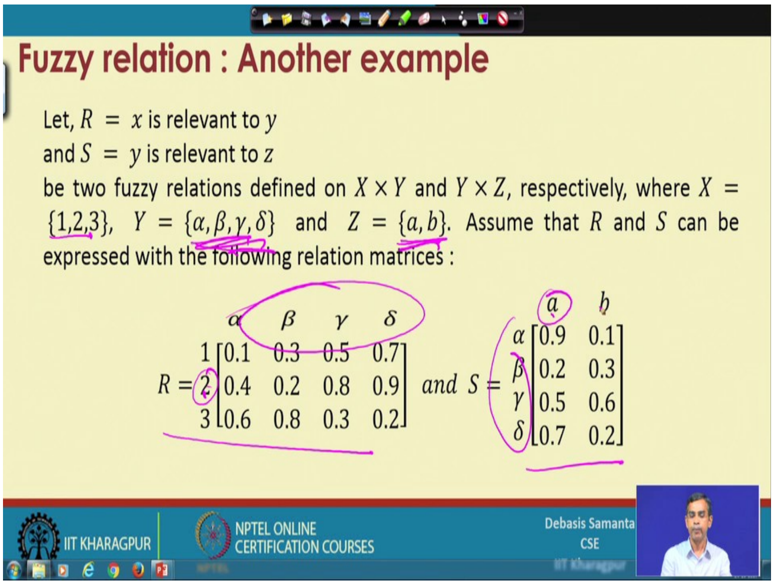
(Refer Slide Time: 27:34)



Say, S also another relation. It basically showing the relation about disease and the symptoms. So here different disease are there and different symptoms and this is the matrix showing how particular disease and related to the different symptoms for that. Now having this one then we can have the composition operation. So, T is a set resultant which is basically $R \circ S$. So, $R \circ S$ can be obtain previously we have discuss about R and this is the S and taking the max min composition then you can try and you can check that this is the relation that can be obtained.

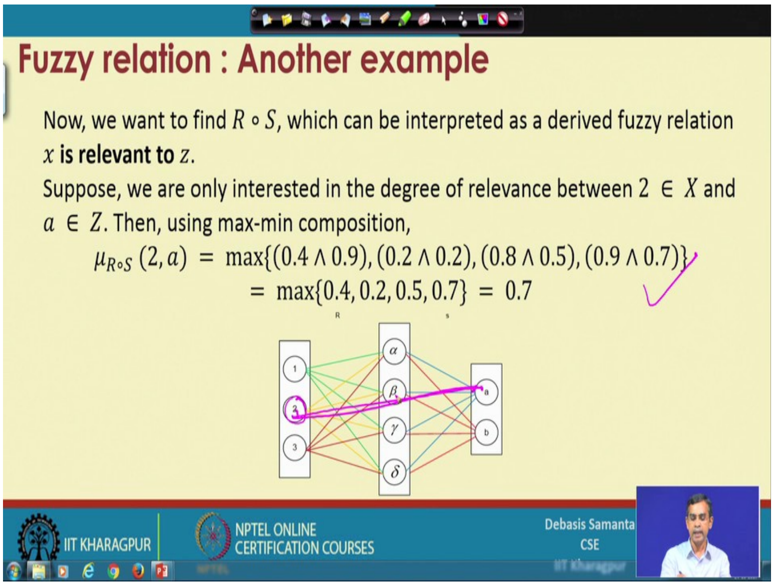
So, this relation has the meaning. This meaning is that if the R relation shows that which paddy plant is susceptible to which disease and if S denotes the particular disease and what are the symptoms, then $R \circ S$ basically shows a particular plant then what are the symptoms that it basically corresponding to some disease. So, this is the relation showing a plant and then symptoms that they may be affected. So, this is the one example and we hope I hope you have understood the concept of relation.

(Refer Slide Time: 28:49)



And this is the another example that is the I can give it very quickly, so that you can understand. If R is another relation showing the relation from these sets to these sets. Sorry. This is the R relation is this sets to this set and another relation S showing the relation from these element to this element. If given this one then I can find a relation from any to anyone, so via this one. So, α, β, γ basically in between the two and then we can find the relation to $2a$ or relation 2¿b . So, that can be obtained by means of Cartesian product and then relation composition operation rather not Cartesian product.

(Refer Slide Time: 29:30)



So, it can be like this one. For example, $2a$, 2 is the element which is belongs to set and these are the different and a. So, you can find the relation from $2¿a$ by means of max min composition or the max min composition is can be calculated which is shown here.

So, it basically here. For example, 2 and a there is a relation via other elements α, β, γ. So, the relation that 2 is related to a which strength 7 0.7. Similarly we can calculate likewise the 2 and a relation between 2 and a. We can calculate relation between 1 and b or 1 and a with some what is called the strength. So, this basically shows the relation and it is the meaning it is there. Now, so I think time is over. So, we can stop it here. These portion can be discussed in the next lectures.

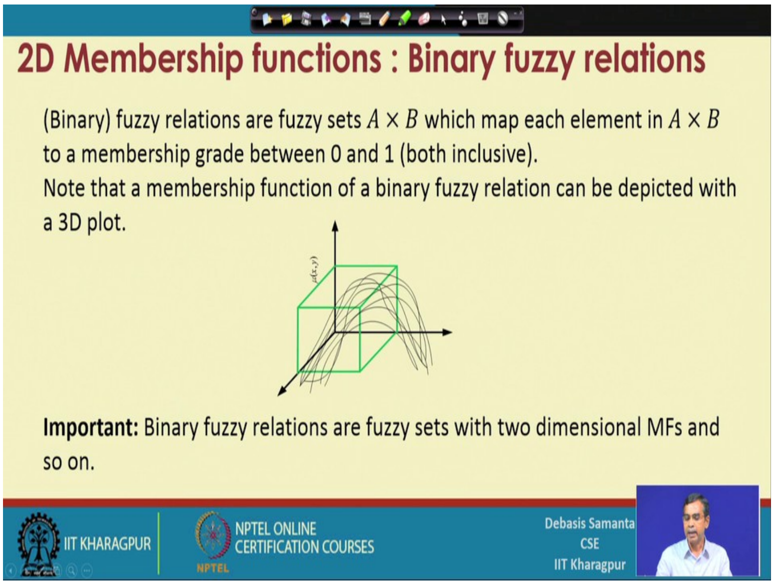
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 06
Fuzzy Relations (Contd.) & Fuzzy propositions

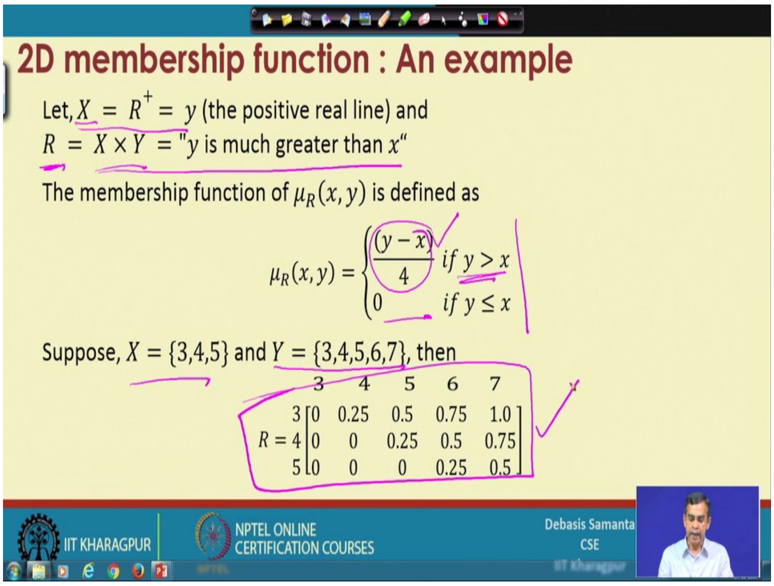
In the last lecture we are discussing about fuzzy relation and there are few portions are remaining. So, I will start with that remaining portion and then finally, I will go to the another concept of fuzzy it is called the fuzzy proposition. So, the fuzzy relation that we have discussed as we see it is called the binary fuzzy relation. Binary means it is the relation over the two fuzzy relations. So, that is why binary or the union, intersection or the composition they are binary fuzzy relation other than the complement that is the intersection it is basic, the complement of the fuzzy relation is a unary fuzzy relation.

(Refer Slide Time: 00:57)



Now such a fuzzy relation can be also graphically display and that graphically it is look like. So, $\mu(x, y)$ is basically the membership values of the relation and if we plot in a I mean a 3D graphs then the graphically representation it will look like this is this kind of form like; that means, the membership function how varies. So, the graphical representation can be plotted with some mathematical tools like MATLAB if the two relations given or the fuzzy the sets are given and all these thing. So, sometimes the mathematical operations and using some tools we can pictorially describe different relation or operation to understand how system works.

(Refer Slide Time: 01:48)

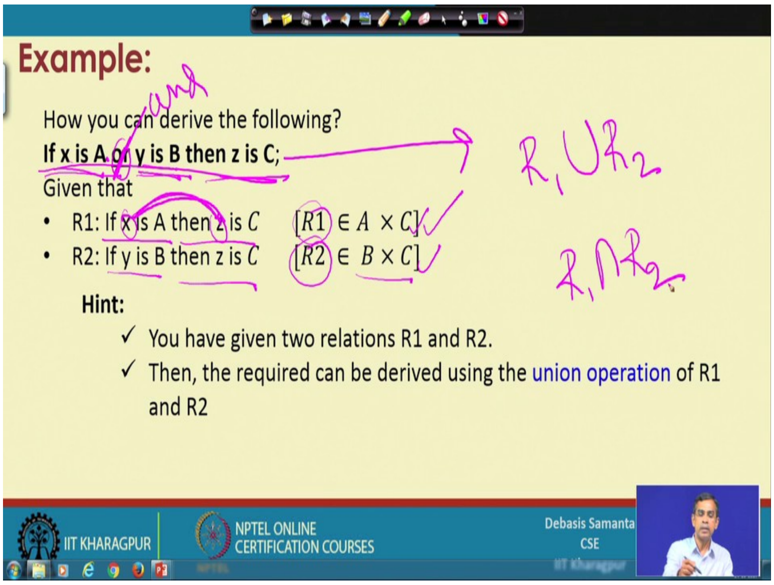


Now I want to discuss about this binary fuzzy relation with some mathematical meaning in it and here is an example that I want to show. This example you can check it carefully. So, X and Y, X is a fuzzy set which is defined like this over the universe of discourse $\overset{+¿}{R^{¿}}$.

It is basically positive values or positive numbers and R is the another relation which is defined over the discourse X and $X \times Y$ or Y is also another relation is defined over the a set of positive numbers. And the relation X is basically or Y it is the relation is defined here that "y is much greater than x". The membership function of such a relation can be defined more mathematically using this formula suppose. So, is a much greater than it is discussed by these relation and if it is y is greater than x and if y less than x then it is 0.

So, this basically gives the membership value for a relation be between x and y is an order pair in the relation and it can be depicted in the form of a matrix. Now here an example, so x is a set y is another set and if we follow this relation then the relation matrix that can be obtained like this one. So, this is basically is an another way of representing 2D membership function is a binary function also other than the max min composition that we have discussed. So, there is many interpretation that can be applied, and then fuzzy relation can be obtained; now after this fuzzy relation is known to us.

(Refer Slide Time: 03:37)



This is the another implication of the fuzzy relation. So, basically $A \times B$ if it is a fuzzy relation as you know that is basically say the relation that if it belongs to a particular relation then what is the strength.

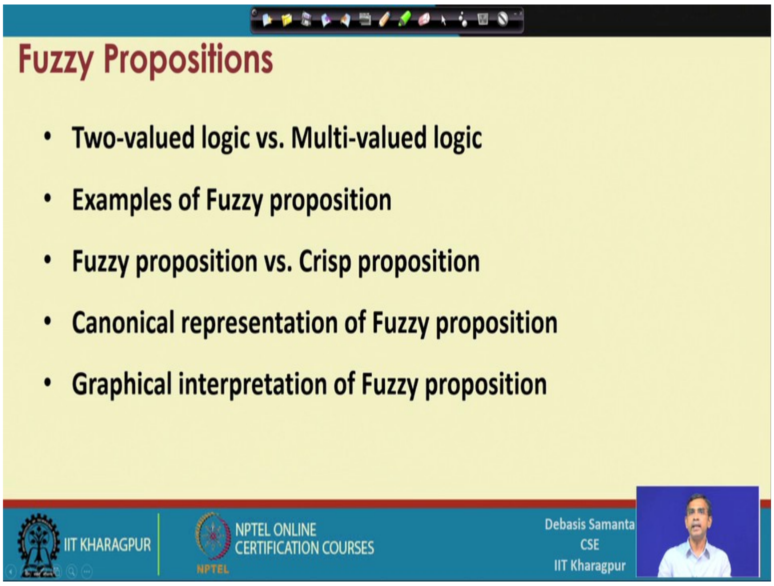
Now, I can say one example here. See if **x is A or y is B then z is C**. So, this kind of concept can be expressed by means of two product like. See if **x is A then z is C** it can be represented by means of the product Cartesian product of the two set A and C. So, this gives the relation **R1**, similarly **if y is B then z is C** it can be represented by the Cartesian product B and C and this is the relation **R2**. Now a Cartesian product between the two fuzzy sets results a relation and a relation takes the form like this if **x is A then z is C** this means that, how x and z is related or the relationship strength between x and z which belongs to the set A and set C likewise.

Now, if here this is another example if you see. If two relations are related then what will be the result in this one. For example, if **x is A or y is B then z is C** how it can be represented. So, it is basically if x is A this is another relation y is B then another relation. So, it gives z is the another relation. So, if the two relations are given to you by this form, so this kind of expression can be obtained, that this is the relation R1 or is a R2 so union. Similarly if I say if x is A instead of **or** it is **and** then that relation can be obtained by intersection this R2.

So, this basically shows some application where the relationship is used and they can be applied. So, the operation that we have discussed there we can apply and then we can find other relations as well as this one. Next we will discuss about another important elements in the fuzzy system it is called the fuzzy proposition. So, exactly what is the proposition? Proposition basically one statement. So, statement like sun rises in the east. So, it is a proposition. What is the truth value? Truth value means sun rises in the east it is value is basically either 0 or 1 or true or false. So, it returns true.

Similarly the many propositions can be given. For example, the mango is sweet. It is a proposition and it can result any value sweet yes no or may be sweet, may not be sweet or these kind of things are there. So, proposition not necessarily give only 2 value either yes or no or true or false, it can give any value. So, in case suppose it gives only 2 values then we say it is a binary proposition or binary proposition is also sometimes called the predicate proposition or simply the crisp proposition. Now the fuzzy proposition means that truth value; that means, the value of a proposition that is possible not initially only 2 values it can be two or more values.

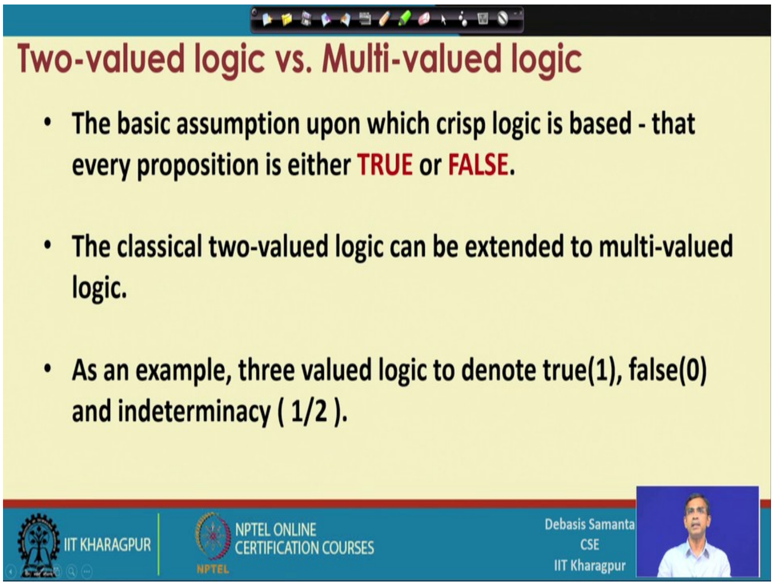
(Refer Slide Time: 07:31)



So, let us see how the fuzzy proposition can be defined and how it can be applied in our fuzzy system and what are the different operations those are possible in order to have the fuzzy proposition, more meaningful implication or significance. Now fuzzy proposition is basically as so fuzzy logic, in fact is called the multi-valued logic in contrast to the Boolean logic is called the two-valued logic. So, this is the one difference between the crisp logic or fuzzy logic or Boolean logic or fuzzy logic. So, crisp logic or Boolean logic they are basically two-valued logic or as the fuzzy logic is basically the multi-valued logic.

So, exactly this is the important difference is there now in this discussion in the coming discussion, we will first learn what is the difference between two-valued logic and then multi-valued logic or how the 2 different logics are related. And then we will discuss about some examples on fuzzy proposition, and then we will try to find the difference between the fuzzy propositions and then crisp proposition mainly the Boolean proposition or predicate proposition. And then we learn about how to represent a fuzzy proposition, this kind of representation is called the canonical representation and the graphical interpretation of a fuzzy proposition.

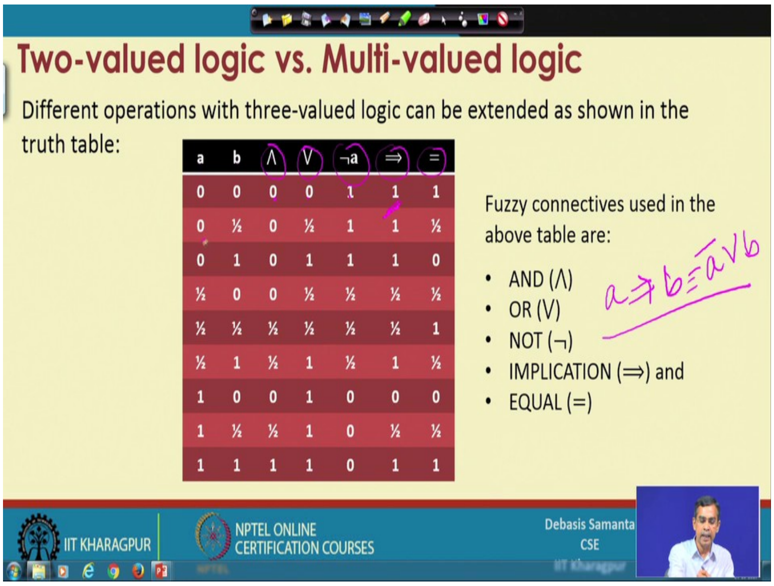
(Refer Slide Time: 09:04)



Then now let us start about the discussion about two-valued logic versus multi-valued logic. As I told you two-valued logic basically based on the two values of the logic. The 2 values are namely true or false sometimes it is represented 1 or 0 yes or no. So, whatever it is there only 2 values. Now the classical 2 valued logic, in fact whatever the concept that is applicable to 2 valued logic can be extended to multi-valued logic. Now it is interesting to note how these extension can be can be take can be taken place in case of multi-valued logic namely say binary logic.

Now, for this illustration we will consider multi-valued logic in terms of 3 values true, false and in indeterminacy; that means, in between 0 and 1 it is say half; that means, a logic can be defined whose values or outcome can be in terms of any 3 value 0 half and 1.

(Refer Slide Time: 10:12)



So, having this is the composition now let us see how the multiple logic can be defined using the conventional two-valued logic operation. Now conventional two-valued logic operation we say it is the $\wedge$ operation $\vee$ operation. $\wedge$ operation is very similar to the intersection operation. The $\vee$ operation is very similar to the union operation. And this is the operation called the complement and this is the implication operation and this is the equal operation.

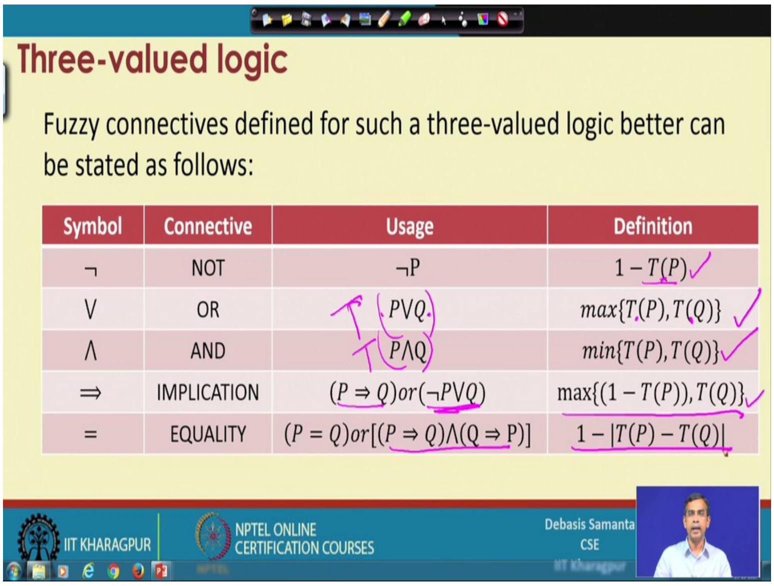
Now if you know Boolean logic then you will be able to understand how the and operation of 2 Boolean variables can be applied or a complement of a Boolean variable can be can be operated. Now here for example, if a and b are the two values for the two variables a and b then it's $\wedge$ operation is 0 and then $\vee$ operation is 0, complement is 1 and this implication is 1. So, implication is basically see $a \Longrightarrow b$ is equivalent to $\acute{a} \vee b$. So, it is the general expression this is basically same thing $a \Longrightarrow b$ is equivalent to $\acute{a} \vee b$. So, this is borrowed from the Boolean logic. Here for example, a complement means 1 or b 0. So, this is why this value is 1.

Now, let us consider. So, these are 0 0 now here 0 and half. Now likewise the $\wedge$ operation is basically 0 and half we have to take the min minimum. So, $\wedge$ operation is minimum.

So, this operation gives the result 0 and $\vee$ operation means taking the maximum 0 and half the maximum is half, so, it is half. And then complement, complement means $\neg a$. So, it is this one if you take the $\neg b$ then you can understand, what is the result $\neg b$ will also result half because is $1-\frac{1}{2}$ and $a \Longrightarrow b$ this and can be obtained using this formula you can obtain this one and $a=b$ it is half this kind of things it is there.

So, these are the table that can be applied for the different operation namely, and which is equivalent to intersection or union is a complementation and implication and equal are the two special operation that is there in case of Boolean logic. Now these are the operation that we can have that this is the I mean truth values for the 3 valued logic and 3 valued logic is very closely related to the concept of multi-valued logic.

(Refer Slide Time: 13:05)



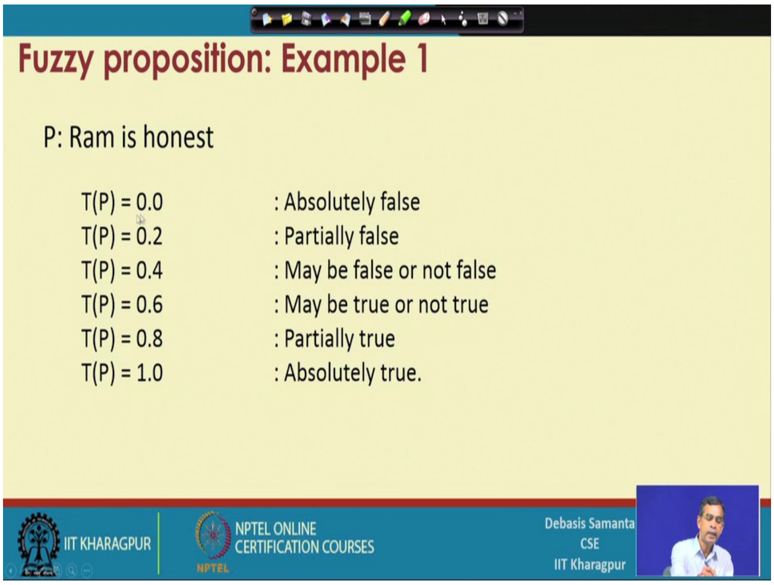
Now, I can little bit move forward to indicate the different operation that can be applied in terms of proposition. Now here for every proposition have some truth value.

If we denote these proposition as a T(P) say suppose P is a proposition, suppose P is a proposition and T(P) denotes it's truth value. Then the complement of this proposition is basically $1-T(P)$. Now or, if 2 proposition P and Q and their truth values T(P) and T(Q) known, then P and Q the union operation of the 2 propositions can be expressed using this formula is a max; that means, $max\{T(P), T(Q)\}$ gives the result of the this is basically T(P) truth value of the P or Q.

Similarly, two proposition P and Q if we apply the intersection operation, which is defined by this expression, then we can give these are truth value of these operation. Then implication it is, I told you $(P \Rightarrow Q)$ equivalently $(\neg P \vee Q)$ this can also be alternatively can be observed that it can give this one. So, it is the $max\{(1-T(P)), T(Q)\}$. Now equality with a $P=Q$ it can be defined alternatively $(P \Rightarrow Q) \wedge (Q \Rightarrow P)$ which alternatively can be obtained using this formula.

So, what I want to say is that the propositions are there, propositions may have 2 valued 3 valued or multiple-valued logic and then their operations, different operations these are the different logical operations, symbols and then whatever the meaning and that can be applied and can be obtained using this kind of definition. Now let us see some example, so that we can understand the concept of proposition more meaning more clearly.

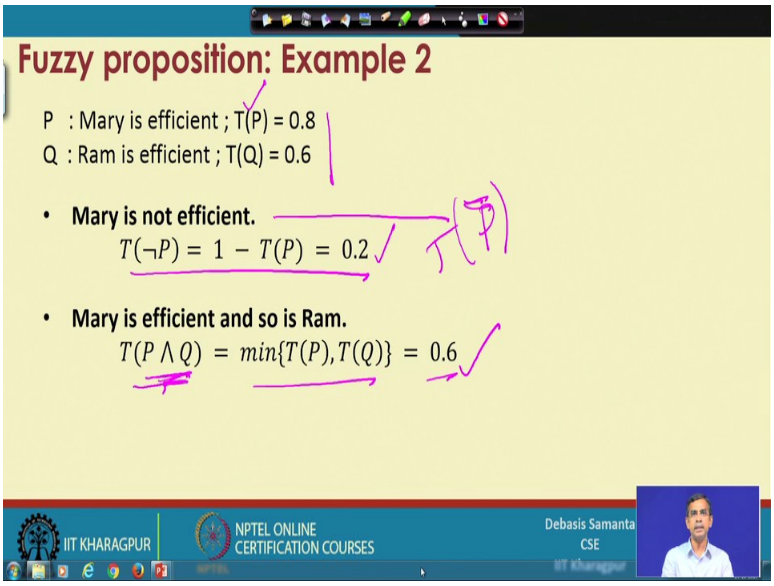
(Refer Slide Time: 15:07)



Say one proposition these Ram is honest. Now, Ram is honest having this proposition can have many truth value. So, T(P) 0.0, it is basically in terms of fuzzy logic; that means, if we draw a fuzzy sets that ram is honest, then the membership function take these value it is like this.

So, according to the different elements it can be obtained like this. So, it is basically the different truth value that is possible, Ram is honest T(P) 0.0, 0.2, 0.4 whatever it is there, a meaning that it is absolutely false is a meaning that it is absolutely false or partially false these kind of things here so; that means, P is a proposition and this proposition may have different values. That we can say the different truth values rather. So, this way it is multiple-valued concepts.

(Refer Slide Time: 16:13)



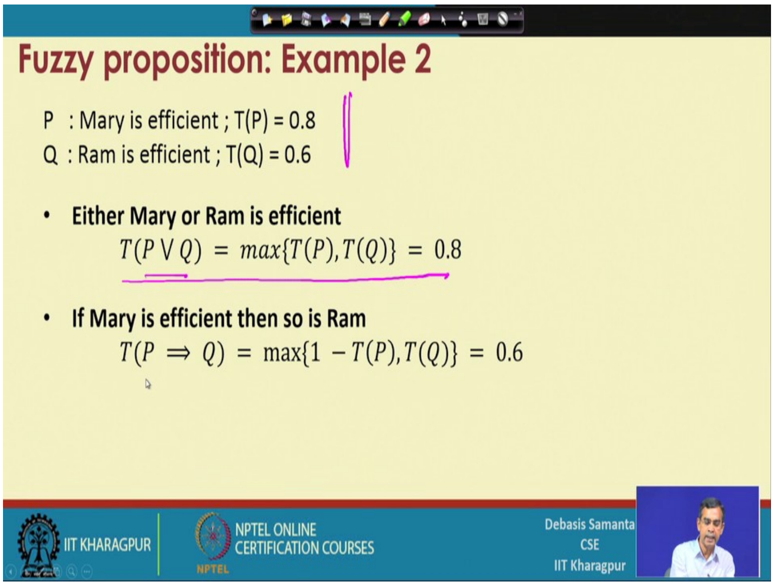
Now, let us see some idea about extending this idea. So, say ram is honest as you have discussed about it. Now let us consider other 2 propositions P and Q, where the P is defined as Mary is efficient and another proposition Q it is defined as ram is efficient. Now this P being a proposition let it's truth value that is denoted as T(P) and it is 0.8, the another proposition Q having truth value T(Q) and is denoted as 0.6. Now these are the 2 propositions, then we can see how the different operations are applicable to this proposition to give something more meaningful proposition.

Now P is the Mary is efficient we can have another proposition Mary is not efficient, this basically equivalent to P and Q are the 2 proposition as you have discussed here with this notation. And Mary is not efficient this is the another proposition and this proposition I can express as a $\acute{P}$ or $\neg P$ whatever it is so P complement. Then if Mary is not Mary is efficient given this T(P) then the T(P) of this $T(\neg P)$ this can be expressed by this formula $1-T(P);$ that means, if Mary is efficient is a proposition then Mary is not efficient another proposition not P having the truth value 0.2.

Now, if we say the another proposition in terms of P and Q are given propositions. Let the proposition be Mary is efficient and so is Ram; that means, it is Mary is efficient as well as ram is efficient. So, if so in terms of P and Q we can apply the and operation then we can obtain the truth values of this new proposition P and Q and that can be obtained taking the minimum values of this one. So, it is basically 0.8 and 0.6. So, this is the results or resultant values of the new proposition $T(P \wedge Q)$ and truth value is 0.6.

So, the different propositions and having the different proposition the different operation that can be applied, those operations very much similar to the operation that is there in Boolean logic that can be further extended to the multi-valued logic that we have discussed using min, using max and some mathematical relation and then we can obtain the different propositions given the some elementary proposition or some basic proposition it is there.

(Refer Slide Time: 18:59)

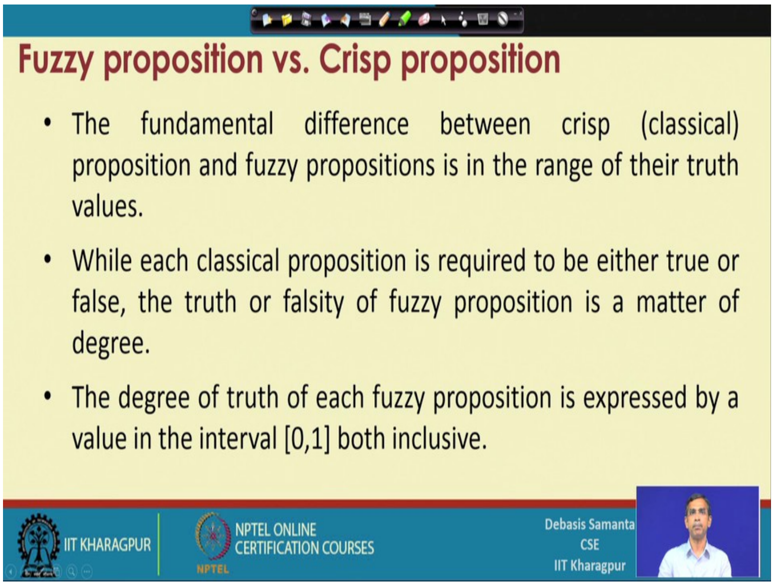


Now again let us see so P and Q are the 2 proposition which is discussed here and then we can have another proposition either Mary or ram is efficient.

So, if it is it is basically represent by means of the operation $P \vee Q$; that means, Mary efficient or Ram is efficient, then truth value can be obtained using the max operation. Another example it is here if Mary is efficient then so is Ram. So, the operation that is applicable to this fuzzy set is basically is basically represented here if Mary is efficient then so is Ram. So, we can represent this one; that means, if P is true then Q and it is a implication now truth value of this $T(P \Longrightarrow Q)$ can be obtained using this expression $max\{1-T(P), T(Q)\}$ and this value can be obtained at 0.6; that means, if Mary is efficient then so is ram is another proposition with truth value is this one.

So, it is basically in terms of proposition and using some proposition we can obtain some other proposition applying the operations on the multiple logic that we have discussed.

(Refer Slide Time: 20:25)

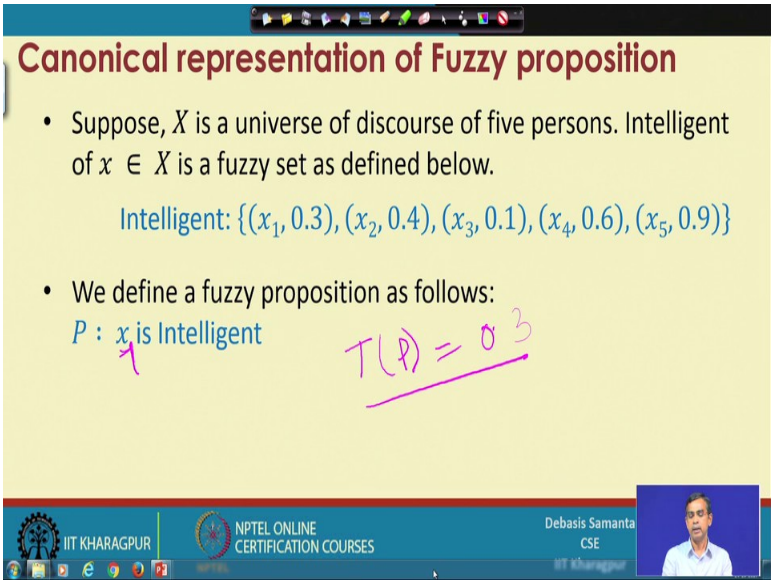


Now, we have learned about the fuzzy proposition and the crisp proposition. The fuzzy proposition basically is the multi-valued logic proposition where the crisp proposition is basically two-valued logic proposition. So, so is a this is the crisp value is also called the classical proposition and the fuzzy proposition is a newly defined proposition for the fuzzy logic.

Now, so here we can see in case of crisp proposition this is required either true or false value to be return, in case of fuzzy proposition it can return any value in between 0 and 1 and this value basically signifying the degree of strength in the propositions the resultant proposition.

So, degree of truth of is fuzzy proposition is expressed in the range 0 to 1 both inclusive. So, this is the difference between the fuzzy proposition and crisp proposition. In case of crisp proposition the result is always in terms of 0 and 1 and in case of fuzzy proposition it is in between any value 0 and 1 inclusive both the things.

(Refer Slide Time: 21:23)

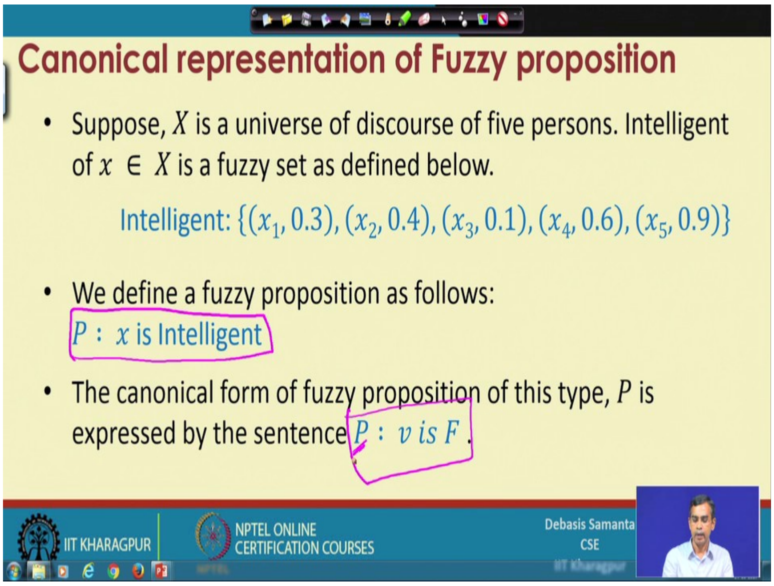


Now, we will come to the discussion about canonical representation of fuzzy proposition. Because whenever we learn about fuzzy logic we have to express some canonical way the universal way. To do this thing let us consider the X is a universe of discourse. This discourse consist of 5 persons and we want to define one fuzzy proposition that intelligent. So, X is intelligent like say, ram is honest with the different value. So, intelligent is a fuzzy concept or is a fuzzy linguistic and any element X that can be belongs to this fuzzy set intelligent can be defined using this formula this is the fuzzy set.

So, now here you can check it. So, x_1 is a one person I say that he is intelligent with this degree 0.3, similarly x_4 is a another person he will be treated as a intelligent with degree 0.6. So, this is the fuzzy set defined over a discourse universe of discourse of 5 person and their membership value is represented here. Now having this representation now I can say that we can define a fuzzy proposition in terms of these fuzzy sets like this: x is intelligent. Now here x is basically can be qualified any one of this one x_1 or x_2 or x_3 this one; that means, then it will return say suppose x is for an example x is x_3 then what is the T(P).

So, it basically T(P) in case of x_1 is intelligent. So, T(P) will give you sorry it is T(P) if x_1 is intelligent it will give you 0.3. So, this is the proposition that x_1 is intelligent and it basically gives the truth value that 0.3. Now so this is the truth value if it is represented, then we can say this basically is a one way of representing a proposition and this is called the canonical representation.

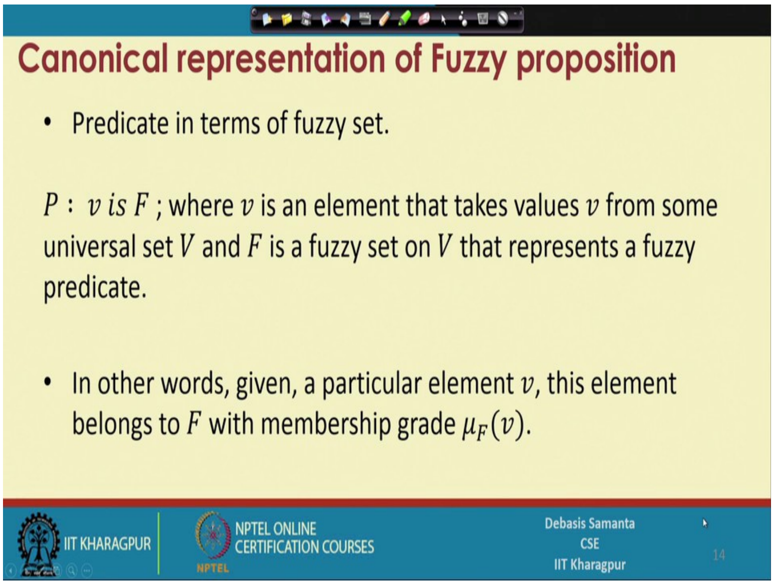
(Refer Slide Time: 24:01)



So, canonical representation in general can be expressed like this form, where P is a proposition and v is an element which belongs to a fuzzy set F and it gives basically proposition that $v \text{ is } F$ and then P gives the proposition.

For example F denotes a temperature and v is say 20 degree. So, 20 degree is so F denotes a hot temperature. So, 20 degree is hot temperature then it basically is a proposition and the 20 degree is a hot temperature whether it is with certain value it can be any value in between 0 and 1, it is basically gives the value of this proposition. So, this is the canonical representation of the fuzzy sets and it is denoted by P as P that is $v \text{ is } F$.

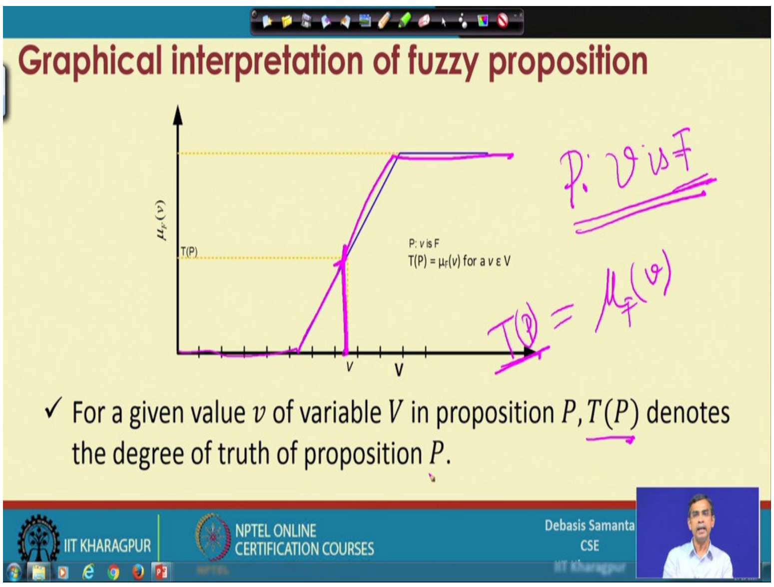
(Refer Slide Time: 25:04)



Now, this is basically other way P in in terms of proposition in fuzzy logic. This P is called the predicate in terms of fuzzy sets. So, P is the predicate and this is the proposition expression in the canonical form.

So, where v is an element that takes value v from some universal or some universe of discourse the V and F is basically a fuzzy sets defined over this universe of discourse V. So, this way we can have the fuzzy proposition or a predicates we can say.

(Refer Slide Time: 25:57)

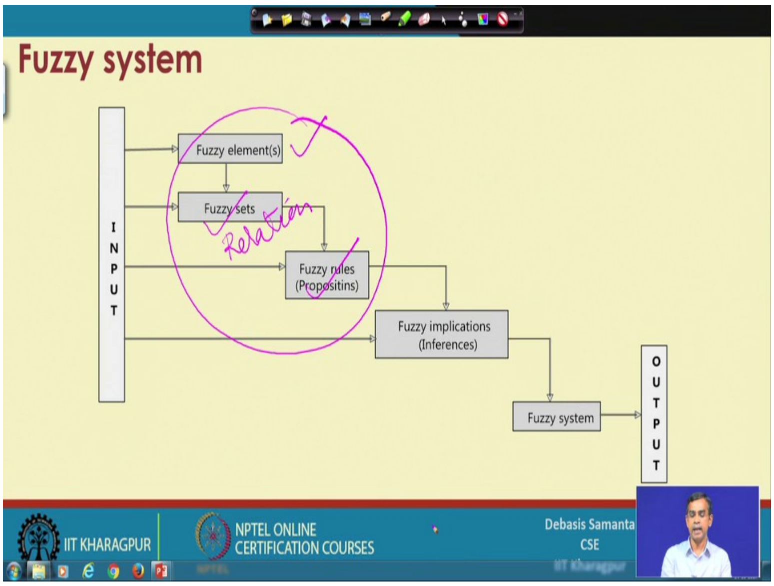


Now, in other words given an given a particular element v this element belongs to F with membership grade $\mu_F(v)$. So, here the idea it is here we can see we can elaborate this concept more meaningfully here and see here. As I told you P is a proposition and it is denoted as $v\ is\ F$; that means, where F is a fuzzy set if F is a fuzzy set then definitely it will be defined by a membership function. This means that the same membership function can be drawn by means of a graph.

Now, here this is the graph of the fuzzy set F; that means, how the value of the membership function varies with different elements and the different elements is defined over the discourse universe of discourse this is the V. Now for any element that belongs to this fuzzy set F let this element is v. Then $v\ is\ F$, it basically gives the membership values for these v. So, it is basically $\mu_F(v)$ now this is basically the T(P) that means truth value of this proposition.

So, it is basically we can write in another way. If P denotes a proposition such that v is F then T(P) basically denotes the membership values for this proposition. So, for a given value v of variable belongs to set V it is in proposition P with degree of value T(P) it denotes the degree of that proposition and this is the graphical interpretation of the fuzzy proposition there.

(Refer Slide Time: 27:52)



So, this is the proposition. Now we are we have discussed many things so far. So, fuzzy element the basic concept we have already discussed there. Now fuzzy elements we have discussed and then we have learned about the fuzzy sets. Fuzzy proposition just we have learnt. In order to learn this fuzzy proposition we have learned that the relations that mean fuzzy sets. So, these are the portion that we have learned so far. Now our next learning objective is fuzzy implication. So, these things will be discussed in our next lecture.

Thank you.

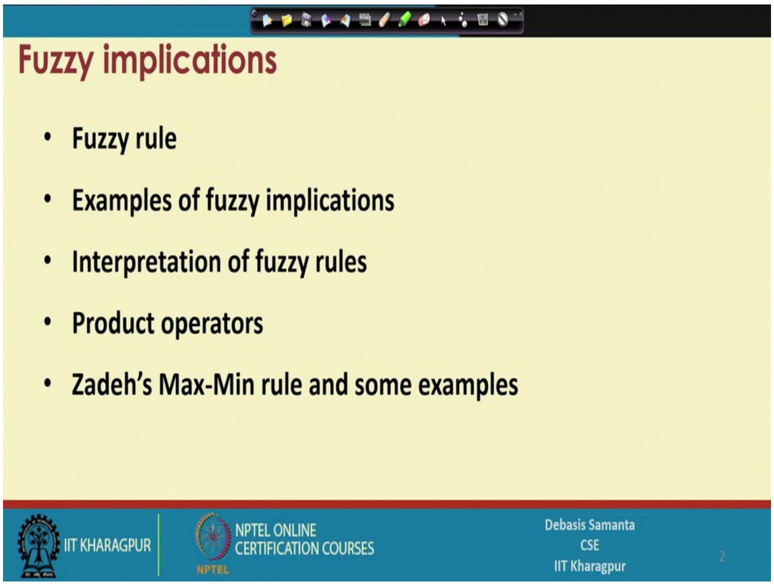
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 07
Fuzzy implications

So far we have learned Fuzzy set. Fuzzy set is the basic building box for the development of fuzzy system. Now in the context of fuzzy sets we have learned different operations so that from two or more fuzzy sets how the other fuzzy sets can be obtained and then you have also learned about the relations between two fuzzy sets to get another, a fuzzy elements.

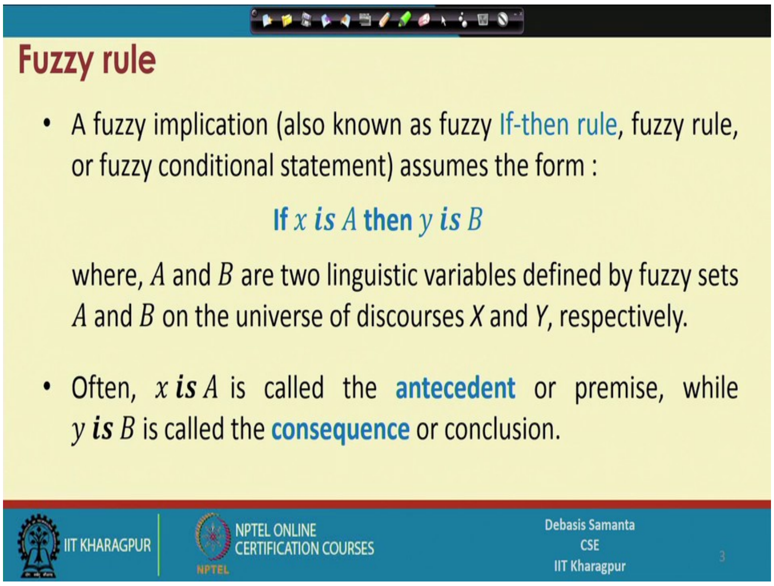
Today we are going to learn about fuzzy implication. Now we know that there is a proposition for a fuzzy set A and which we denote as x is A. So, there is a relation among the propositions. Now such a relation can be better described with the help of fuzzy implication. So, today we are going to learn about fuzzy implication and the different operations or the computation techniques so that the fuzzy implication can be calculated.

(Refer Slide Time: 01:19)



So, today basically we will discuss about fuzzy rules. This is the another name of the fuzzy implication in fact and some examples and then interpretation of these and then some operations by which the fuzzy implication can be computed and one operations which is extensively used in fuzzy system development, it is called the Zadeh's Max-Min rule. We will discuss in details about a Zadeh's Max-Min and we will illustrate the Max-Min rule composition technique with some example.

(Refer Slide Time: 01:55)

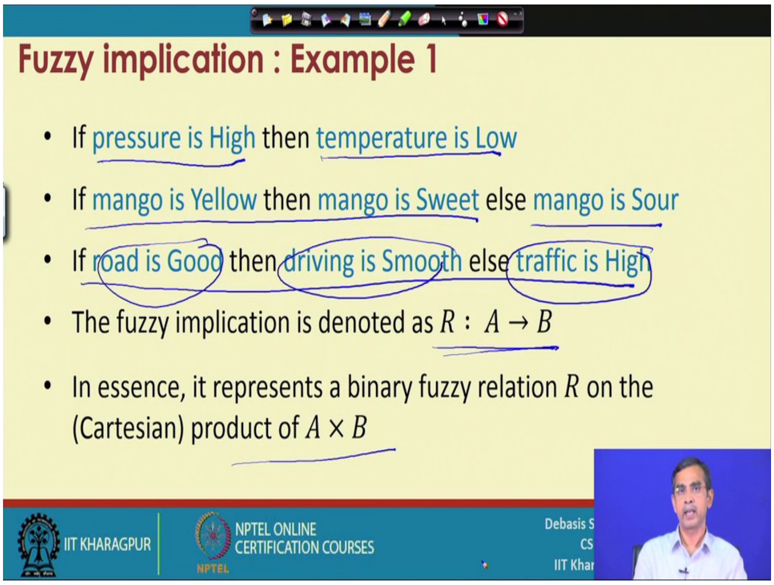


So, now let us come to the fuzzy implication that is also called fuzzy rule. It is also sometime called as if-then rule, sometime it is called a fuzzy conditional statement. Now such a fuzzy implication is basically is a relation between one proposition to another proposition. For example, see this is one example of a fuzzy proposition fuzzy implication and here $x \text{ is } A$ as we know it denotes a proposition; proposition regarding and element x in a set A and another proposition $y \text{ is } B$. So, it basically gives a relation among that two. This is the proposition and this is a proposition and then relation in the form of if then. So, that is why it is called if then implication or if then rule.

Now, it is it is necessary to calculate what it does mean; that means, $if\ x \text{ is } A\ then\ y \text{ is } B$ what value it does return to us. So, we will see that $if\ x \text{ is } A\ then\ y \text{ is } B$ can be expressed in the form of a relation matrix, that mean the fuzzy implication can be stored in the form of a matrix and it is important to calculate the different entries in the matrix. We probably you can recall $if\ x \text{ is } A\ then\ y \text{ is } B$ we represent such a implication or rule in the form of a Cartesian product. But Cartesian product is basically the very basic things in this relation. There are many things that can be considered so in order to represent a fuzzy implication that will learn it.

Now, in this context so fuzzy implication takes in a general form as I told you $if\ x\, is\ A\ then\ y\ is\ B$. Here the first part, this is called antecedent or it is called the premise. And then the this part, the next part is called the consequence or conclusion. So, this is the general form of a fuzzy rule.

(Refer Slide Time: 04:15)



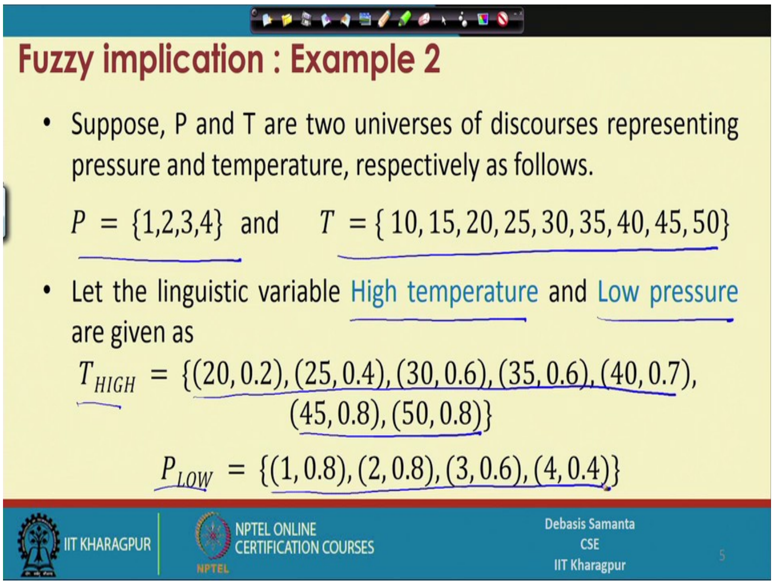
And we are to calculate the fuzzy rule basically the different membership values that can be that can be applicable to such a rule, anyway that will be discussing details in due time.

Now, I can put some examples, so that we can understand how fuzzy implication or fuzzy rule looks like. Here I have given some example, **if pressure is high then temperature is low**. So, as you know, so pressure is high is a another what is called this is the premise or antecedent and this is the consequence, in fact, these are the two propositions.

As an another example, **if mango is yellow then mango is sweet else mango**. It is basically expressed in if then else. So, if then else is another form of course, but sometime we can represent everything in the form of a only if then and this is a unification method usually followed in a fuzzy system. So, anyway, but if then else also another structure that can be represented only in terms of if then else that will be we will discuss it about later on.

As an another example, this is another, **if road is good then driving is smooth traffic is high**. So, here basically relation among 3 prepositions road is good, driving is smooth and then traffic is high. So, the relation takes the form as I have discussed here with in terms of few examples. So far that notation of representing a fuzzy implication is concerned it is usually represented using this form. So R, R represents a fuzzy rule and if it is basically $if\ x\ is\ A\ then\ y\ is\ B$ it is represented $A \rightarrow B$. So, that is why it is also called fuzzy implication and I told you such a fuzzy implication is basically a relation. Relation from the fuzzy set A to fuzzy set B and broadly it can be expressed in the form of a Cartesian product. So, it is in a broad sense it is a Cartesian product actually, but there are some methods by which this relation can be calculated in a more mathematical way that will be a more formal way rather and we will discuss. And we are going to learn this the method by which the fuzzy relation can be calculated.

(Refer Slide Time: 06:41)

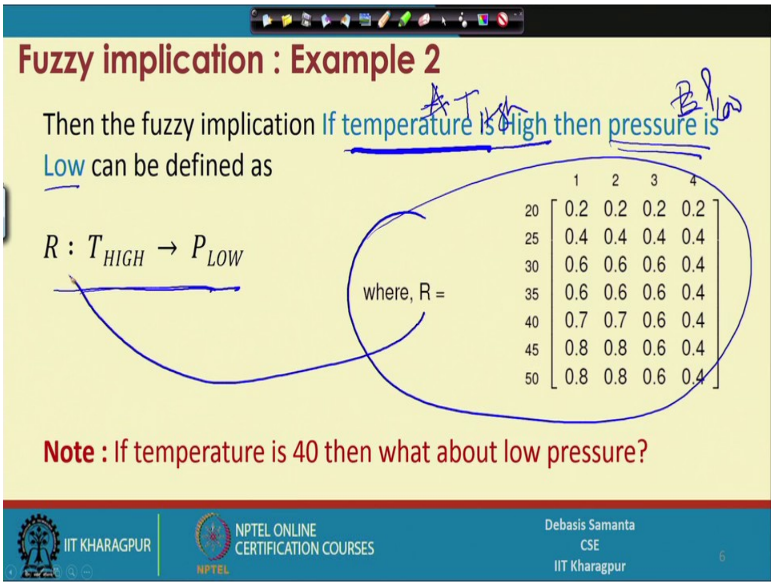


Now let us see one another example. So, that we can understand the fuzzy implication better. In this example, let us consider two fuzzy sets P and T. So, this the P and T, these are the basically denotes that pressure in some scale and temperature in some scale. We can consider pressure and the temperature are the two universe of discourse, giving the value of pressure that is possible and temperature that is possible.

Now, let us consider the two fuzzy sets. They are expressed in terms of linguistic variable called the high temperature and low pressure. So, these are the two fuzzy sets and the two fuzzy sets say high temperature we denote it as T_{HIGH} and the degree of membership values for the different element is shown here. Like likewise so the low pressure it is denoted as P_{LOW} and is represented in the form of a fuzzy set which is shown here.

So, given these are the two fuzzy sets now let us see how the fuzzy implication can be can be expressed.

(Refer Slide Time: 07:55)

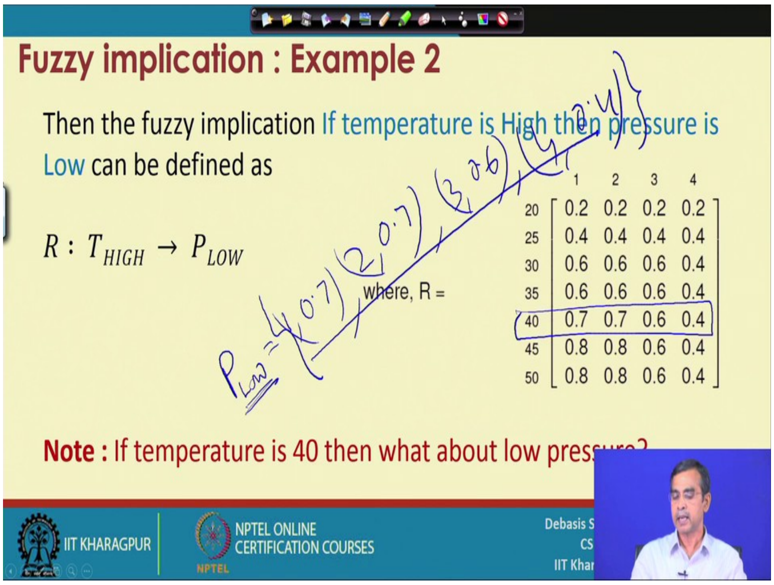


Now, so fuzzy implication fuzzy implication from the two prepositions, say temperature is high then pressure is low. So, here the proposition is temperature is high then pressure is low. So, it is basically this is the one we can say this is A temperature is high and the pressure is B say B. So, it is basically we can express these in terms of like this one. Sorry it is not high it is A we have denoted as T_{HIGH} and this is B here P_{LOW}.

So, the relation or we can say implication between the two can be expressed we using a short form $T_{HIGH} \rightarrow P_{LOW}$. Now using simple Cartesian product that we have already discussed in the previous lecture, we can obtain the relation matrix; that means, showing these it look like this. So, this is a relation matrix R showing this relation R, so using the Cartesian product that we know.

Now, one point one interesting point is that here if temperature is high then pressure is low now say suppose temperature is forty then what about the pressure. So, it is basically if temperature is 40 then it basically shows the pressure.

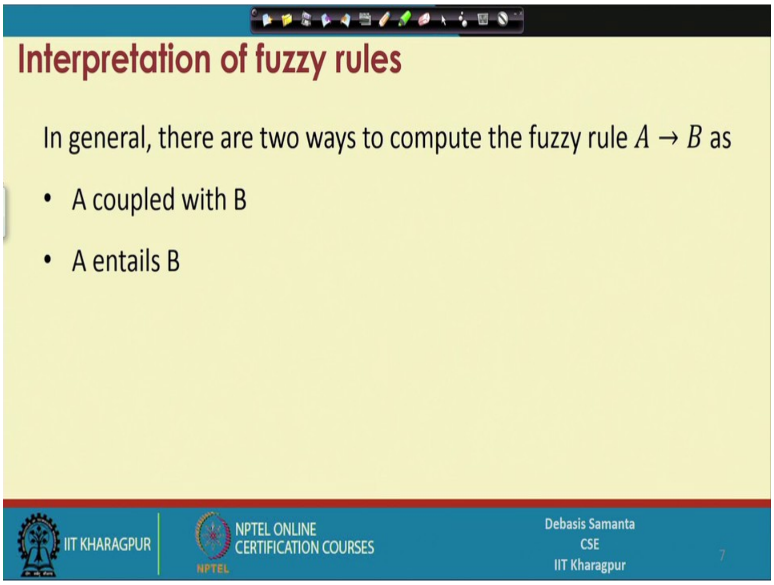
(Refer Slide Time: 09:25)



Now, you can see what temperature 40 the pressure. So, I can say P_{LOW} here P_{LOW} this is basically can be expressed in term of a fuzzy set (1, 0.7) then (2, 0.7) then (3, 0.6) and (4, 0.4). Now, so this is basically P_{LOW} provided that provided that temperature high as 40. So, this basically is the fuzzy set. So, answer is like this. So, this rule gives an answer about the pressure that is pressure as low for a given temperature this one.

So, this is basically one implication or purpose of the fuzzy set that we will use in in our fuzzy system development. So, these two examples can be helpful for us to understand how the fuzzy implication works for us.

(Refer Slide Time: 10:28)

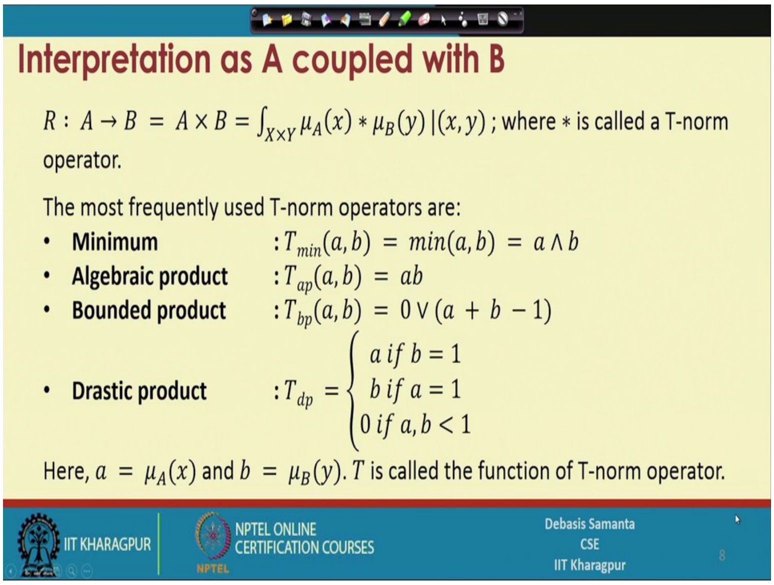


So, now, what is the important concept here it is that, here in this calculation of these relation R, we have used min max simply a min formula, right. If $T_{HIGH} \rightarrow P_{LOW}$, so the Cartesian product and taking the min we obtain these two these are the different entries. But in a more what is called the sense more practical manner there are many other calculations are involved so that this relation matrix can be calculated.

Now, whatever the methods are there they can be broadly classified into two broad categories. One is called A coupled with B and A entails B. Now these are the basically different techniques or different principles to obtain the relation matrix. So, these not necessarily give the same result, because different method different principle or different interpretation give the different results actually, but all those results work in a different context of course, but we can use any one method to I mean calculate our relation matrix. So, it is depends on the fuzzy designer, fuzzy engineer who wants to use in it is system that which method he should consider.

Anyway, let us see what are the different methods are there which belongs to A coupled with B and A entails B.

(Refer Slide Time: 12:02)



Now, let us first start with A coupled with B. So, A coupled with B are usually the relation R in this form and it is basically $A \times B$ instead of only min operation that we have already discussed, so it basically takes the general form which is expressed here. So, the relation matrix it is basically it is a $\mu_A(x)*.$ ¿ it is an operator is an operator among the two membership values $\mu_A(x)*\mu_B(y)¿(x,y)$ that is there in the relation. So, here this ¿ is called specifically it is called an operator, it is called the T-norm operator.

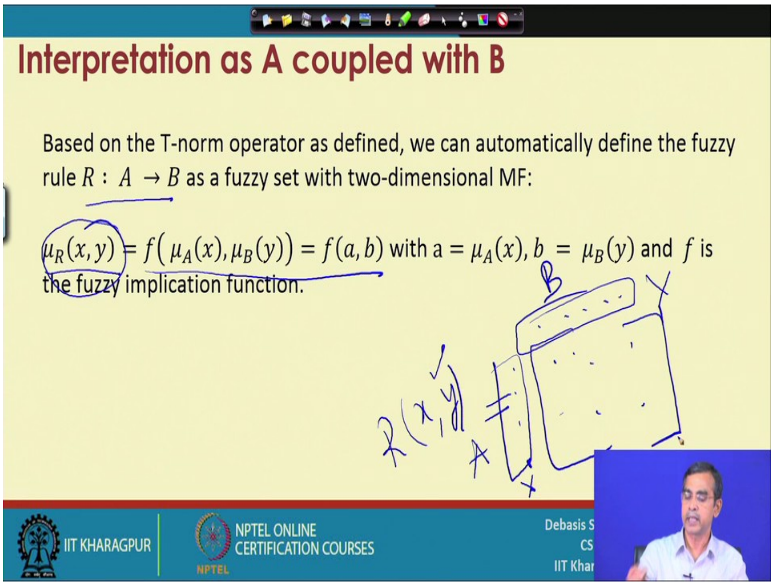
Now, let us see what are the operations that this T-norm operator can signify. Here we see 4 different operations that can be can be explained or that can be applied so far the T-norm operators are concerns. So, basically operators is symbol ¿ it can be applied to compute either these or these or these one. So, different operator differ same operators can perform different calculations or different results depending on what type of operation that we are fixing for a relation in a relation.

Now, the first operation the T-norm operator it is called the minimum and if you see $T_{min}(a,b)$ this is the form this is a $min(a,b)$. So, if $A \times B$ and if it is the $min(\mu_A, \mu_B)$ that is basically the general Cartesian product we have learned so far. So, T_{min} that is a minimum T-norm operator is our the usual Cartesian product that we have already learned.

Now, there is other it is called the algebraic product. Algebraic product is expressed is a product of the two membership values. Now like algebraic product it is called the bounded product. Bounded product is defined using this expression it basically takes the value of it is a maximum of 0 or this 1. So, it will take the value.

Drastic product it is denoted as T_{dp} . So, ¿ can be consider T_{dp} and if we follow this one in $a \, if \, b=1$, it will take the value b if $a=1$ and it will take the value 0, if $a,b<1$. So, what I can say is that for the different entries we can have the different value if we follow the different operations according the T-norm operator. So, if we follow min then it will give another relation matrix, if we follow algebraic product it will give another relation matrix, bounded product another. And if we follow this kind of operation then it will give another relation. So, relation matrix will vary if we follow the different operations. So, if a particular operation is followed it will give a unique value. So, it is the concept that is the A coupled with B the operation.

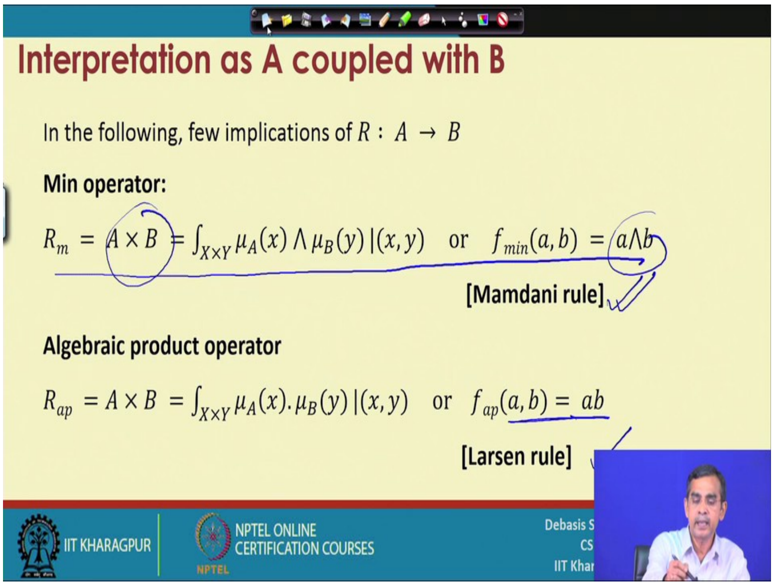
(Refer Slide Time: 15:07)



Now, so based on this T-norm operators right, we can automatically define the fuzzy rule that is $R:A \rightarrow B$, we can define a fuzzy rule $R:A \rightarrow B$. So, this fuzzy rule can be as we told you that this fuzzy rule can be expressed in terms of a two dimensional membership function. A two dimensional membership function is basically $\mu_R(x,y)$ this is basically the value this one. And representation of these two dimensional membership function is basically takes in the form of a matrix and usually this is related term as $R(x,y)$. So, this matrix is this one, where x is any element in this direction and y is any element in this direction.

So, x an element particularly belongs to a particular universe of discourse and y is an element belongs to a particular discourse. So, it is basically universe of discourse of y and these are the element of discourse x. So, these entry whatever it is there it can be calculated applying the two fuzzy sets A and B that is from the two universe of discourse x and y and following T-norm operator.

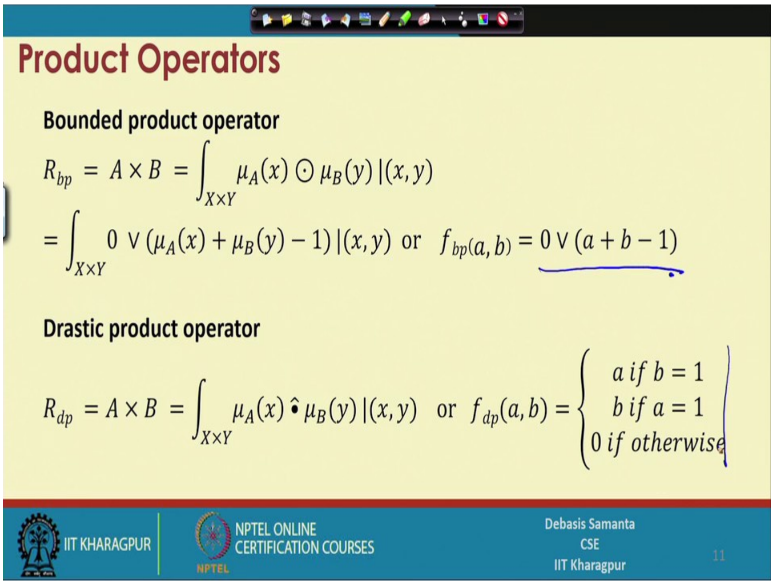
(Refer Slide Time: 16:26)



So, this is the basic concept of calculating the fuzzy relation calculation of fuzzy rules. Now interpretation of A coupled with B now I will come to the interpretation. So min operator as we have discussed already, so min operator T_{min} it basically takes this form. Now such a rule is popularly known as Mamdani rule. If we follow T_{min} and in fact the Cartesian product that we have discussed in last lectures we follow this $A \times B$ as a min and then this one it basically follow the Mamdani rule there.

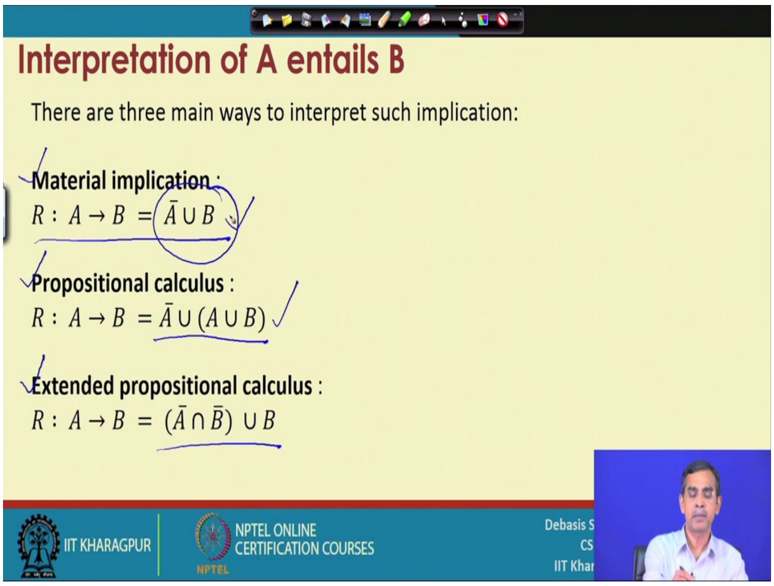
Now, algebraic product operator just we have discussed about just simply a product then it basically call the Larsen rule. So, the difference scientist who has proposed these rules according to name all those I mean rules name like Mamdani rule and then Larsen rule.

(Refer Slide Time: 17:16)



There are some other rules also bounded product operator. It is just simply there is no rule as such specific like Mamdani rule or Larsen rule, it is just expressed using this formula using this formula, in this formula. And drastic product operator as we have discussed using this one, actually no specific in I mean rule or the name we assign to this kind of things are there. They are basically the different way to calculate the fuzzy implication.

(Refer Slide Time: 17:49)

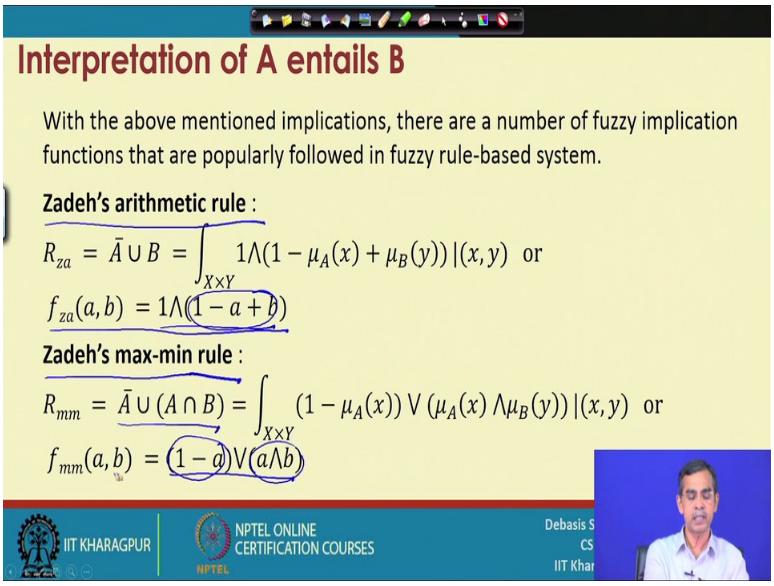


Now, I will come to another type it is called A entails B. Now A entails B takes 3 forms in fact. The 3 forms are called material implication, the propositional calculus form and another is called extended propositional calculation calculus form.

Now, according to the material implication a rule a fuzzy rule is described in the form like this one. So, fuzzy rule is here is basically $\acute{A} \cup B$. So, we know how to perform this operation $\acute{A} \cup B$. So, this rule can be obtained like this. Now the propositional calculus takes slightly different from which is shown here. It is basically $\acute{A} \cup (A \cup B)$ also take this form and the extended propositional calculus take this form.

Now, what we can learn from all these thing all these discussion is that, so the different way of calculating the relation matrix, that's all. Whether these things are equivalent not that is a another question. What I say that these may be this is and this is not necessary the equivalent. This means that if you follow these and if we follow these they are not necessary give the same relation matrix. The different relation matrix interpretation may be different and then purpose or the application may be different. So, that is the thing and regarding application it is not necessary; it is not the right time to discuss; when we discuss about it is a application when we discuss about this meaning that what are the different context the different rule that can be the different way the rule can be calculated is applicable.

(Refer Slide Time: 19:39)

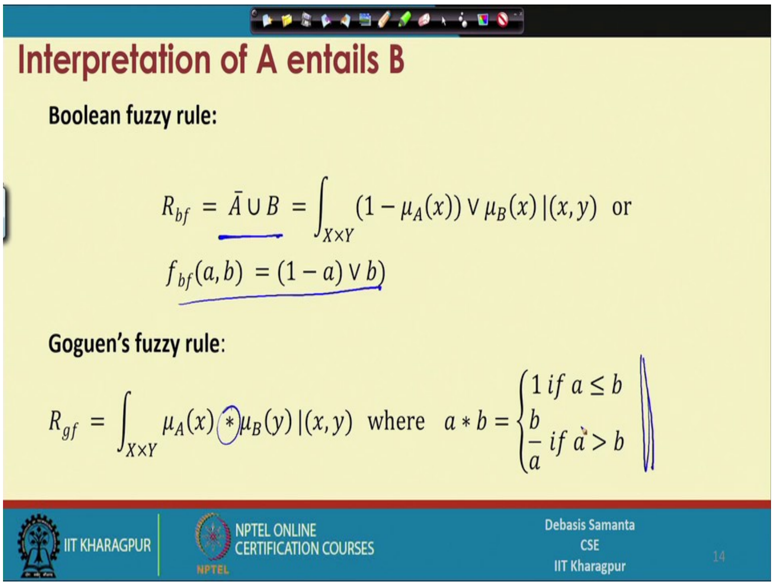


Now, so A entails B needs certain more discussion about how they work for us. Now I will go one by one to discuss each and everything. So, the first rule it is called the Zadeh's arithmetic rule that we have discuss a material implication in fact. So this Zadeh's implication is basically the material implication, it is according to the Zadeh's name it is called the Zadeh's arithmetic rule. And Zadeh's arithmetic rule, that means, if the two values A and B two membership values A and B given, then the relation entries for these two values can be obtained according to this formula is a basically $1 \wedge$ that means, it is basically min of $1 \wedge$ the min of the result of this one $(1-a+b)$.

Now, another implication that is basically the propositional calculus propositional calculus. This rule is again proposed by Zadeh's and it is called the Zadeh's Max-Min rule and it is notation is like this. So, it basically express in this form. It is basically max of $(1-a) \vee (a \wedge b)$. So, max of $(1-a) \vee (a \wedge b)$ so this way. So, this basically the method or calculation to find an entry for two different values there and this one.

So, these this is basically the two rules which is in the techniques of called A entails B.

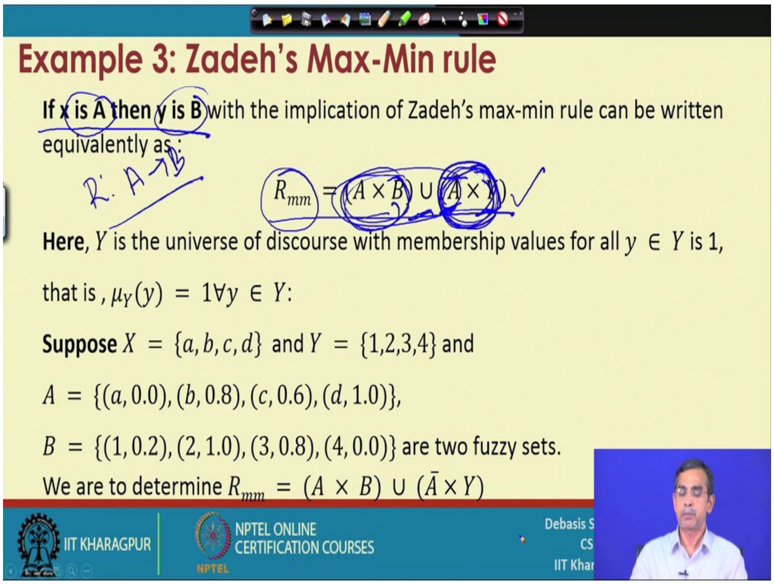
(Refer Slide Time: 21:03)



And there is another rules also belongs to A entails B, it is called the Booleans fuzzy rule. Boolean fuzzy rule takes these form. We can see this is already expressed the material implication according the Zadeh's min rule and here arithmetic min rule, but here in this context Boolean fuzzy rule they have given the different interpretation $(1-a) \vee b$.

Similarly, the Goguen's fuzzy rule is a operation is like this and it is defined by this formula this expression. So, this is also another way to calculate the relation matrix.

(Refer Slide Time: 21:40)



Now, here I want to discuss more elaborately about Zadeh's Max-Min rule this is the one what is called the method which is frequently used or more fuzzy engineer prefers this rule to calculate their relation matrix regarding a rule computation.

Now, let us see what exactly the Max-Min rule it is basically Zadeh's Max-Min rule that just now we have discussed it, but it takes slightly different form. I will tell you why this form is anyway. So, this is basically to calculate the rule **If x is A then y is B**. So, these also can be say the R from A to B where A this is a proposition related to A and proposition related to B. So, this is the rule $R: A \rightarrow B$.

Now, so far the relation matrix is concerned and this can be better expressed in the form $(A \times B) \cup (\acute{A} \times Y)$, where Y; Y is the universe of discourse for the set B and A is a fuzzy set B is another fuzzy set. So, if we use this formula then we will be able to calculate the relation matrix which basically represents the rule like $R: A \rightarrow B$ where **if x is A then y is B**.

Now, you can recall when we discussed about Zadeh's Max-Min rule we wrote that it is basically $R_{mm} \acute{A} \cup (A \times B)$ only this part. But here to make it generalized, we follow this one right. Now the because if $A \times B$ is basically can be store this result in the form of a 2 dimensional matrix and then $\cup A$; however, A is a 1 dimensional matrix. So, it is not possible to apply the union operation on these and these one.

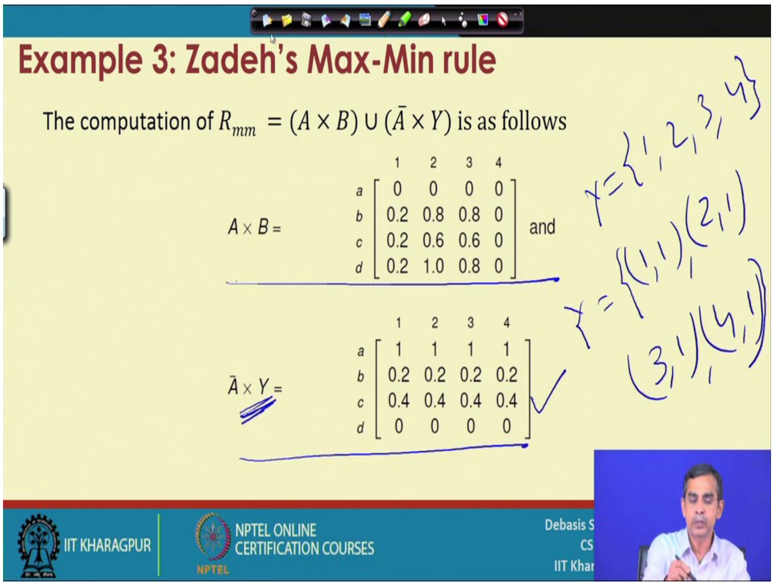
Now, in order to make it applicable so a another cross Cartesian product is applied over the $\acute{A} \times Y$. So, then it gives a relation matrix it gives a relation matrix 2 matrix are of same size and therefore, union operation can be applied; however, if we use this one result can be a bit different, but that is absolutely not an issue. Because so far the certain fuzzy is concerned this result is acceptable.

Now, so this is the concept regarding the Zadeh's Max-Min rule, now let us elaborate this Max-Min rule technique to compute the rule matrix using an example. So, this example I would like to refer here, say X is a universe of discourse and Y is another universe of discourse. This universe of discourse contains 4 element $\{a, b, c, d\}$ and it contains 4 elements $\{1,2,3,4\}$.

Now, let us consider these are the 2 fuzzy sets which are defined over X and Y respectively. So, A is defined over X and B is defined over Y. Now we want to calculate the relation matrix, that means **if x is A then y is B** form and this basically in this form. So, you want to calculate this one.

Now, let us see how this value can be calculated. So, the idea is very simple we have to calculate first $A \times B$ and then $\acute{A} \times Y$ and then take $\cup$. So, the relation matrix a can be calculated. Now here is an idea about the details calculation, so that we can understand about it.

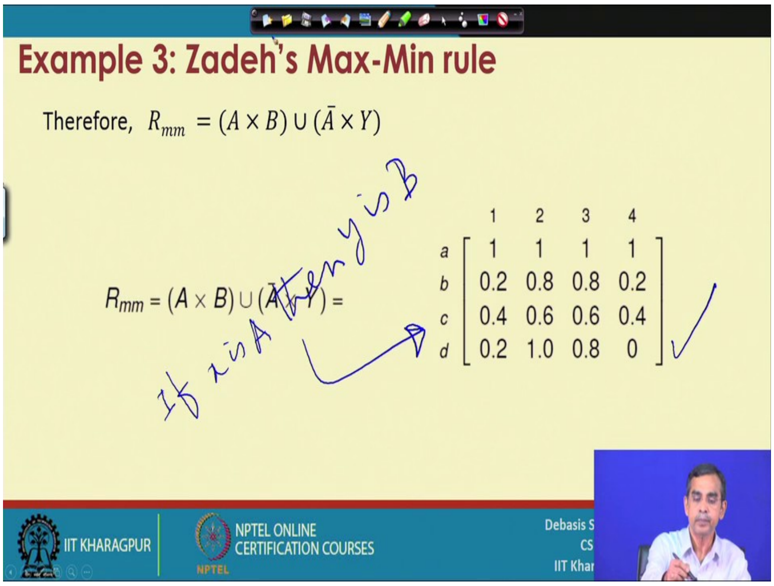
(Refer Slide Time: 25:22)



So you see so given the 2 fuzzy sets A and B you can check that $A \times B$ give this 1 and $\acute{A} \times Y$ takes this form. Now here whenever we consider Y, so Y is basically for all the discourse element that is belongs to the universe of discourse with their membership value one. So, Y is basically {1, 2, 3 and 4} and when we compute this result we take that Y this is equals to (1,1) because in the fuzzy form (2,1) (3,1) and (4,1).

Then we can take the product Cartesian product $\acute{A} \times Y$ taking the min of these one and this kind of matrix can be obtained. I hope you have understood these things whether how these can be calculated. Once $A \times B$ and $\acute{A} \times Y$ is known to us we will able to obtain the final matrix that is basically a relation matrix.

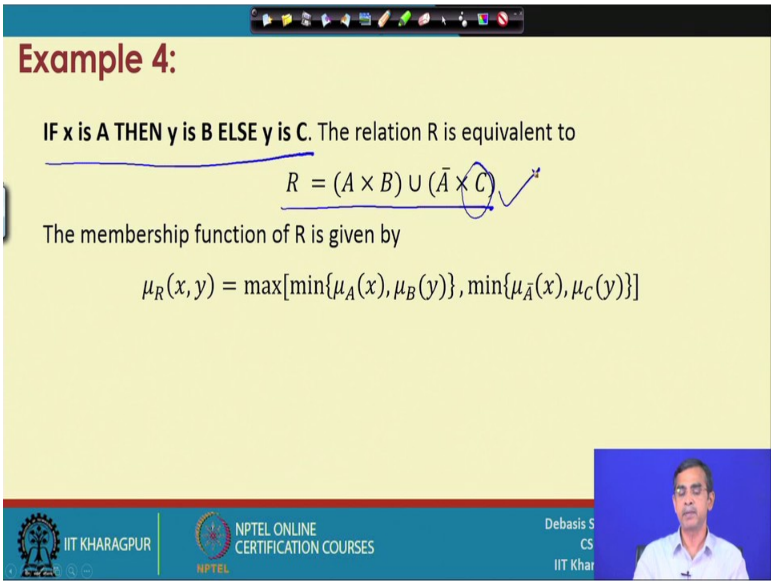
(Refer Slide Time: 26:28)



And the relation matrix will takes this form. So, this basically is a relation matrix represents a rule that rule as I told you that rule is **if x is A then y is B**. So, this basically stored this basically can be represented in the form of a relation matrix.

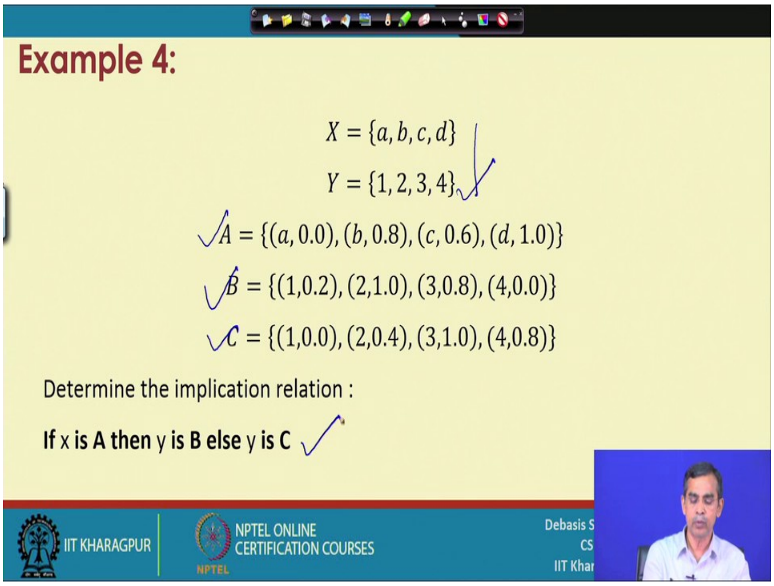
Now, usually we follow in our illustration in subsequent lectures we follow generally Zadeh's Max-Min rule, otherwise you can practice or you can compute another matrix another relation matrix following other operation likes a T-norm operator or some other A coupled with B operation whatever it is.

(Refer Slide Time: 27:27)



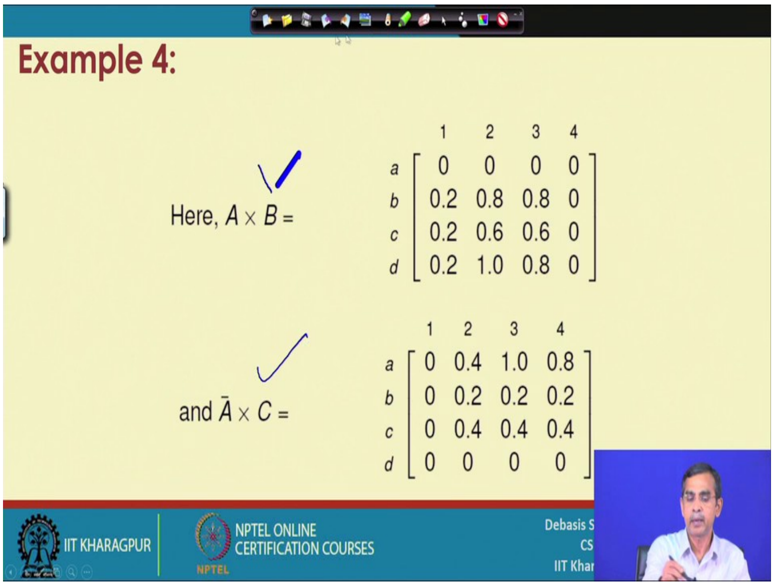
Now, so this basically gives a calculation about a rule **if x is A then y is B**. Now the same Zadeh's Max-Min rule can be extended to another type of rule, here I have mentioned the else part right **if x is A then y is B else y is C**. So, if it is like then it can be expressed using Zadeh's Max-Min rule composition it is like this one. We can note that this part that is a instead of Y we use C. So, this is the way that can be calculated and the rest of the things is very similar to the previous calculation here basically this one. So, this else part is extra else part is extra if else C is added then we can add this one in this Max-Min composition.

(Refer Slide Time: 28:23)



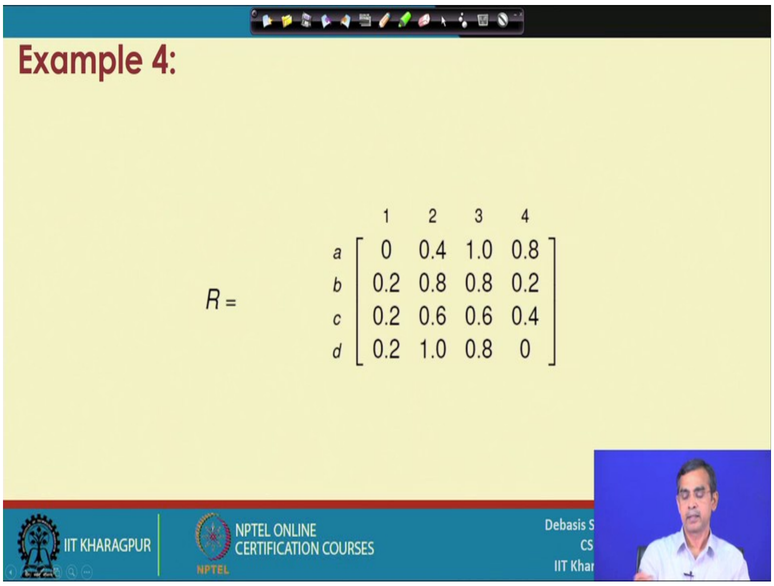
Now, these are example that is basically to calculate this one. So, let us see here X and Y are the 2 universe of discourse defined over the A fuzzy set B fuzzy set is defined over X and Y respectively C is another fuzzy set defined over this universe of discourse.

(Refer Slide Time: 28:49)



Now, this is the rule we can calculate the rule using the Zadeh's Max-Min composition here the $A \times B$ calculation and here $\acute{A} \times C$ calculation. And finally the rule R this basically giving **if x is A then y is B else z is C**.

(Refer Slide Time: 29:02)



So, this is the rule that we can consider for this rule computation.

Now, with this let us conclude here about this fuzzy rule calculation. We have learn many method out of which we want to limit our what is called the process in order to Zadeh's Max-Min method.

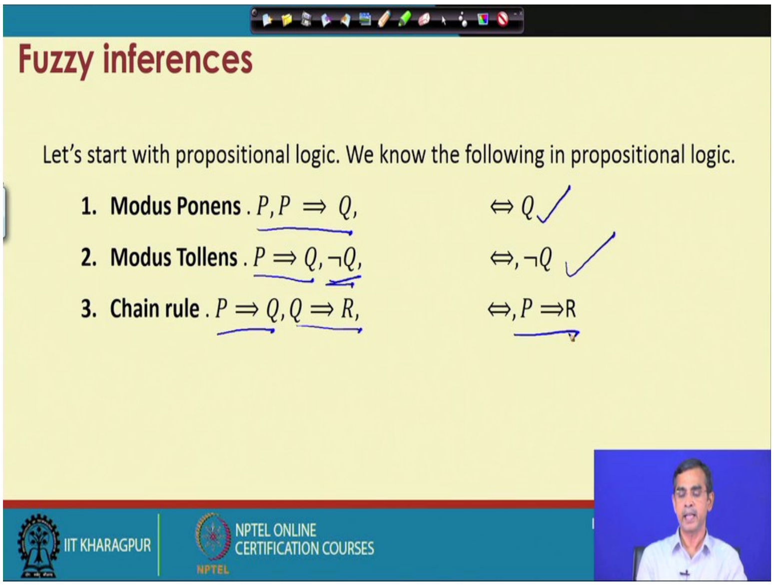
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 08
Fuzzy Inferences

Fine, so we have learned about fuzzy implication. So, fuzzy implication basically calculates fuzzy rule and now we will discuss about how given a set of fuzzy rules, we can infer some other fuzzy rules. So, this this topic is called fuzzy inferences.

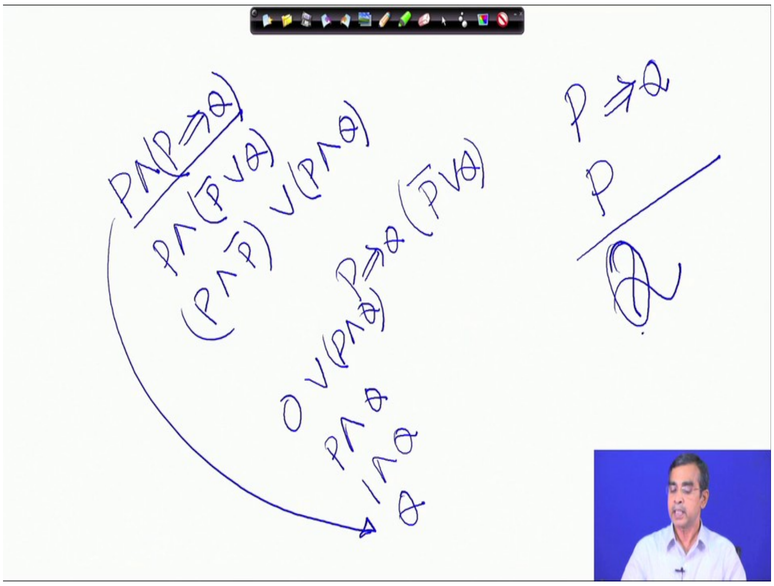
(Refer Slide Time: 00:44)



Now, in order to understand the fuzzy inferences, we need some discussion about which is very much popular there in predicate calculus of predicate logic. Probably we know that operations one is called the Modus Ponen and Modus Tollen. I first discuss above the Modus Ponen then the Modus Tollen can be understood automatically. And then I will discuss about these rules. Actually these are the different what is called the logical rules that can be applied to infer some other rules actually.

So, let us see first the Modus Ponen. The Modus Ponen is a very famous rule that is known in the predicate logic. It is basically the concept about if the two propositions or two formulas it is given to you how can derive another formula. So, this is in the context of predicate logic and we know predicate logic is a two valued logic, but our fuzzy logic is a multi-valued logic only the difference is there, but the most of the method that is the here it also applicable to fuzzy logic or it has been extended to fuzzy logic.

(Refer Slide Time: 01:58)



Now, the idea it is like this, say suppose $P \Longrightarrow Q$. So, this is the one what is called the rule and P is the another rule say it is a proposition or it is a formula whatever it is there. Now the question is that so from this ($P \Longrightarrow Q ¿ \wedge P$ whether we can infer something and if we if we can then how it can be inferred. Now, in the context of this inference we assume that this $P \Longrightarrow Q$ takes the truth value true. That means, it is always 1 or is a true. Similarly, this P also takes the truth value true. That means, the two formula which has the truth value true and the another formula P these are the truth value true. Then how another formula can be derived? So, that it is truth value is true.

So, in case of predicate logic all formula use the truth value true and then false is automatically there. So, it is there. Now let see if such a two rules are given to you how you can derive one rule I can tell simple algebraic manipulation. So, basically ($P \Longrightarrow Q ¿ \wedge P$, I can write the two rule is given P and $P \Longrightarrow Q$ it is like this. So, the two rules can be combined together to these one.

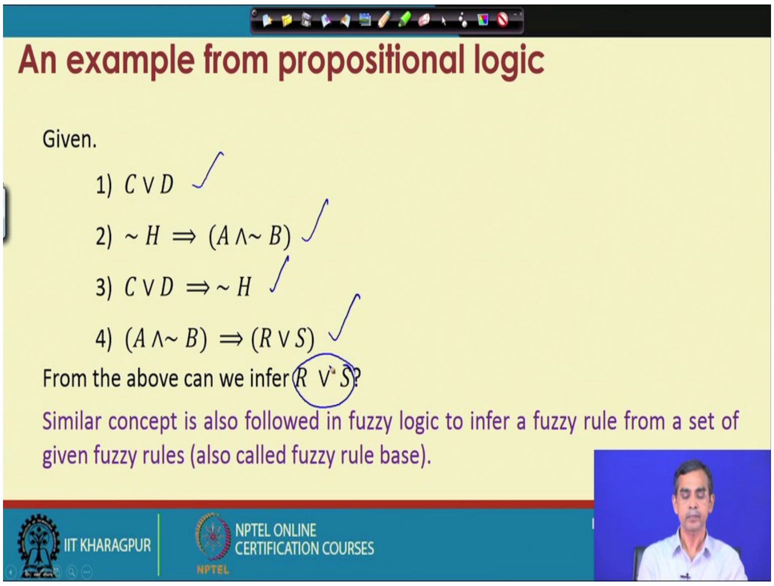
Now, we know that $P \Longrightarrow Q$ is $P \Longrightarrow Q$ also alternatively P written as $\neg P Q$ it is basically $P \Longrightarrow Q$. So, $P \Longrightarrow Q$ also can be expressed in this form. So, using this I can write $P \wedge (\acute{P} \vee Q)$. And using the distribution law, we can write ($P \wedge \acute{P} ¿ \vee (P \wedge Q)$.

Now, $P \wedge \acute{P}$ always give the results 0 so 0. This $\vee (P \wedge Q)$. Now 0 $\vee (P \wedge Q)$ implies is basically $P \wedge Q$.

Now, as I told you P always takes the truth value 1. So, I can write $1 \wedge Q$. So, $1 \wedge Q$ means I can write it is Q . So, what I can say that if given these are the two premises then we can derive another premises it is called the Q. So, what I can say $P \Longrightarrow Q$ if it is given and P is given then we can write it a Q. So, these basically is the basic concept that is the formula there in Modus Ponen. Now the Modus Ponen it is like, if P and $P \Longrightarrow Q$ is known then we can conclude Q . Likewise, the Modus Tollen says if $P \Longrightarrow Q$ is true and $\neg Q$ is true then we can imply $\neg Q$. And this is the chain rule if $P \Longrightarrow Q$ is true and $Q \Longrightarrow R$ is true then we can infer $P \Longrightarrow$ R.

So, these are the three rules that we have mention here. There are many such rules in the predicate logic that is not the topic of our discussion here. So, we can use this concept to extend it into the fuzzy rule also. So, that is discussion in fact, in our next hour.

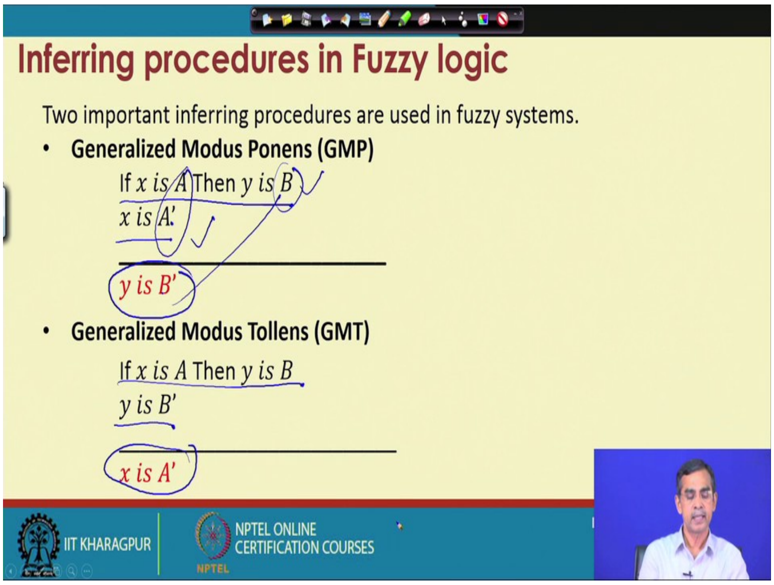
(Refer Slide Time: 05:28)



So, it is basically another example that you can follow here these are the different premises that is given to here this is there.

Now, you can try and you can easily find that from this premises we can conclude about another $R \vee S$. So, the concept is like this. Now, the similar concept is basically applicable to the fuzzy algebra.

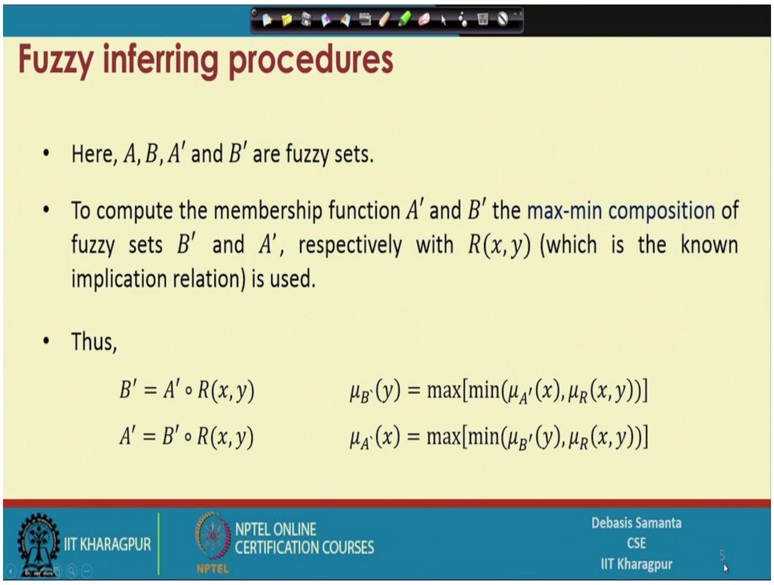
(Refer Slide Time: 05:58)



And, now we are particularly interested about the two such inferences formula they are called Generalized Modus Ponen and it is another is called the Generalized Modus Tollen or GMP and GMT. Now the Generalized Modus Ponen is basically the form it is basically one rule we have already learn about a rule **if x is A then y is B** and this is the another proposition $x \text{ is } A'$, A' is another fuzzy set, where A and B other fuzzy sets and one thing you should note that this A and A' they should be defined over the same universe of discourse where B is the fuzzy set it defines another or similar fuzzy universe of discourse may be, ok.

So, what is the idea is that GMP idea is that if this is given and this is given then we can conclude or we can infer another proposition it is there why it is called the $y \text{ is } B'$ where B' is the another fuzzy set and B' is defined over the same universe of discourse as B. Now similarly **if x is A then y is B** given and here $y \text{ is } B'$ is given then we can infer $x \text{ is } A'$. So, here $x \text{ is } A'$ is given then we can inverse $y \text{ is } B'$. Here $y \text{ is } B'$ is given then we can in can infer $x \text{ is } A'$. So, this is two rules it is popularly called GMP and GMT and using these two rules we can infer some other fuzzy rules.

(Refer Slide Time: 07:39)

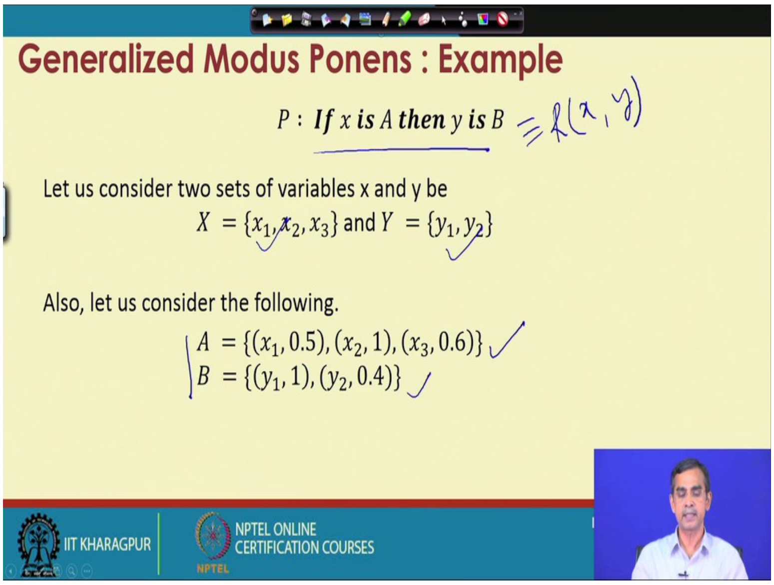


Now, let us see how these rules can be applied to our different fuzzy operations or sets.

Now, so the idea it is like this. So, here basically the input that is given to you two fuzzy sets A, B and then either A' and B' and then you have to conclude either B' and A' from this. Now so this basically can be obtained now you can see the first rule that is there **if x is A then y is B** it can be expressed in the form of a relation matrix and we can represent $R(x,y)$, and the next premises if $x \, is \, A'$ then this relation matrix $R(x,y)$ and then A' can be used and then can be obtained the B' applying the composition formula.

So, it is basically the composition relation composition formula that we have already discuss in previous lectures. So, it is a composition operation. The $\circ$ operation takes this form. Now, on the other hand if B' is given and if rule matrix is given then we can calculate A' using this composition formula. So, it is like this. So, this is the basic idea.

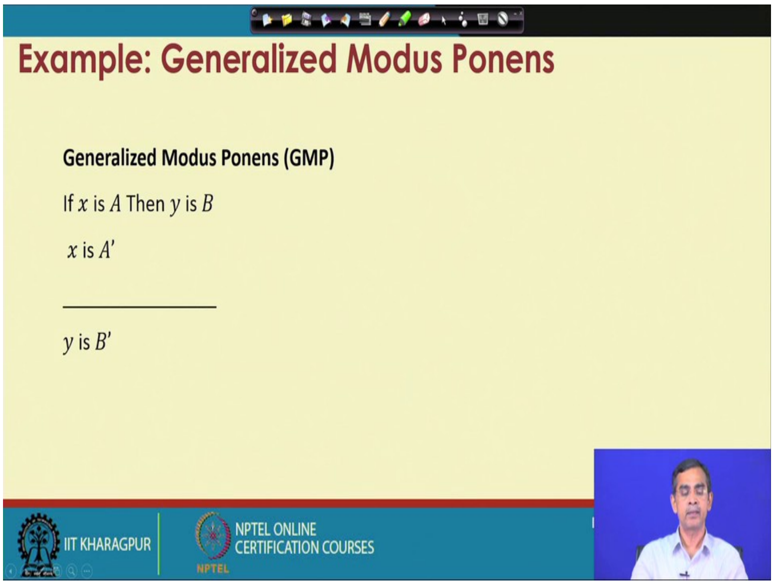
(Refer Slide Time: 08:59)



And let us see how we can use this idea to solve our some problems. So, I want to give an explain illustration about it. Let, this is the one rule that is **if x is A then y is B**. That means, it can be expressed in the form of a relation matrix $R(x,y)$ and this relation matrix can be calculated using Zadeh's Max-Min rule like that ok.

Now, let us consider two fuzzy sets A and B which are defined over the universe of discourse X and B is defined over universe of discourse Y and the two fuzzy sets are given here and here. So, these are the two fuzzy sets. Now given the two fuzzy sets then we will be able to obtain the; we can apply the GMP.

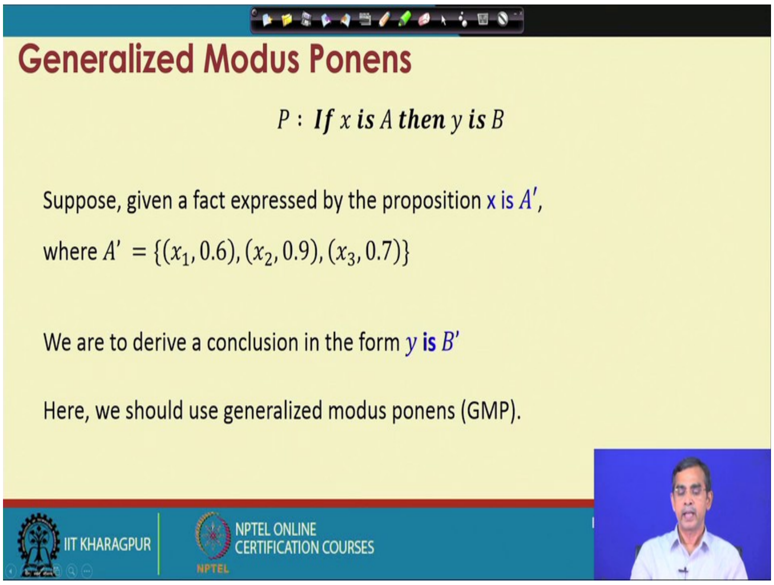
(Refer Slide Time: 09:42)



And then we can conclude about another fuzzy set B'. So in order to do these things. So, this is the GMP we can follow I just want to give an example regarding the GMP another example regarding the GMP also will be given.

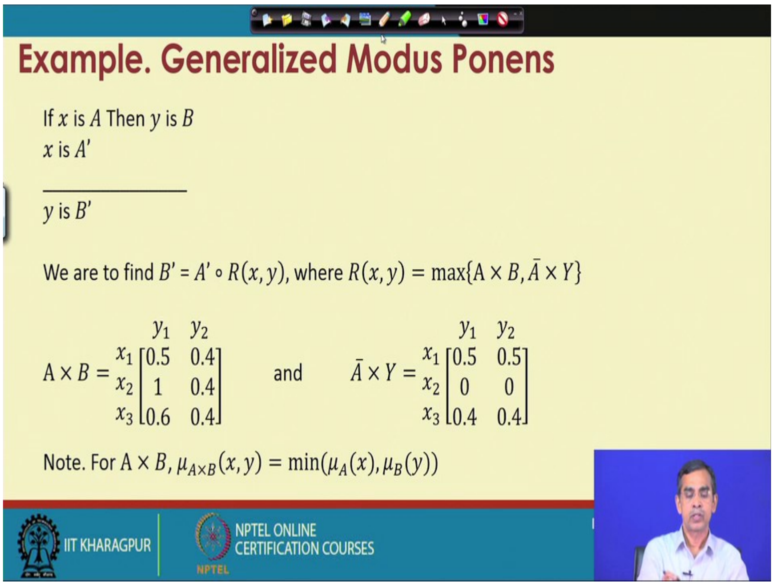
Now, so this is the GMP can be followed if this is given in the form of a $R(x, y)$ and x is A' is given to you then we will be able to calculate B'. Now, let see one example in this direction.

(Refer Slide Time: 10:10)



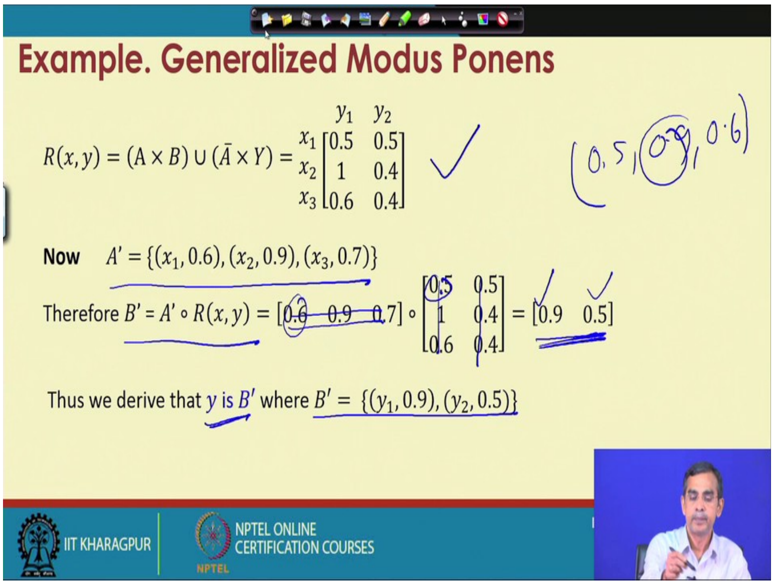
So, x is $A^{'}$; x is A' this takes this form where A' is this one and we are to derive or we have to infer $y \text{ is } B'$ it is like this. Now, so as it is x is A' is given and this is the rule matrix and we have to include it. So, GMP is applicable here now let us see how the GMP can be calculate use and then the relational matrix can be obtained ok.

(Refer Slide Time: 10:35)



So, we have to use this formula as you have discussed it given that $R(x, y)$ is available to us using the Zadeh's Max-Min rule. Now, so if we can apply the Zadeh's Max-Min rule from the given set A and B we can be able to calculate $A \times B$, $\acute{A} \times Y$ and then finally, the rule matrix $R(x, y)$.

(Refer Slide Time: 11:04)

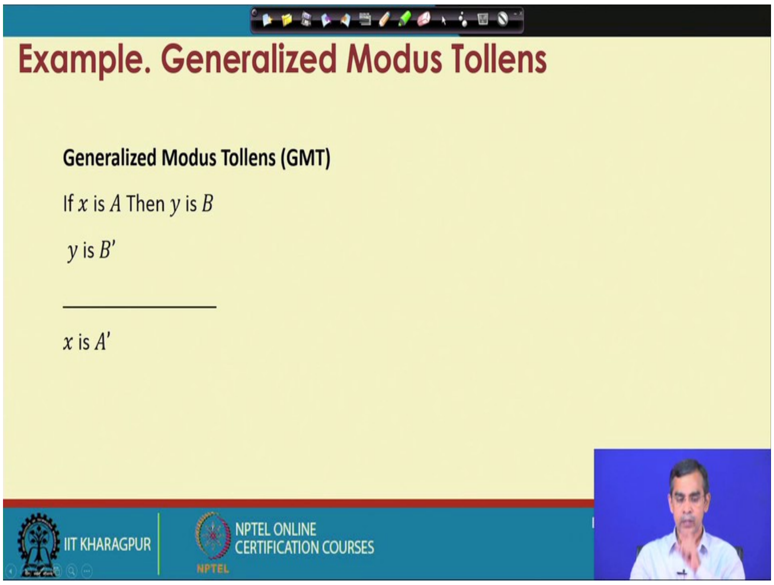


So, here is the rule matrix $R(x,y)$. These are rule matrix $R(x,y)$ that can be obtained given a B and then they are universe of discourses and this is the another input A' is given to us then B' can be calculated like this.

So, it is basically max that is a composition formula again this is a max-min composition formula we know. So, this means we can apply this one take these and these minimum so right. So, 0.6 and 0.5 we can take the minimum 0.5, then 0.9 and 1, we can take the minimum 0.9. 0.7 and 0.6 take the minimum and then take the maximum so 0.9. So, the first entry 0.9, similar if we apply this and this, we can obtain the 0.5.

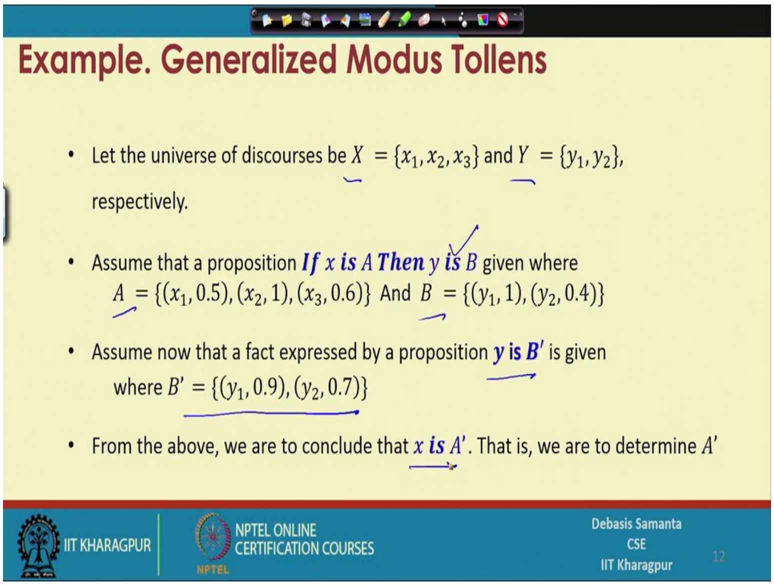
So, B' has the membership values for it is elements 0.9 and 0.5. Alternatively, we can write B' as $\{(y_1, 0.9), (y_2, 0.5)\}$. So, this is the application that we can conclude from a rule another proposition the proposition is that y is B'.

(Refer Slide Time: 12:29)



Now, let us consider another example of GMT it is the same way if we understood the GMP then it is also similarly equally understandable easily.

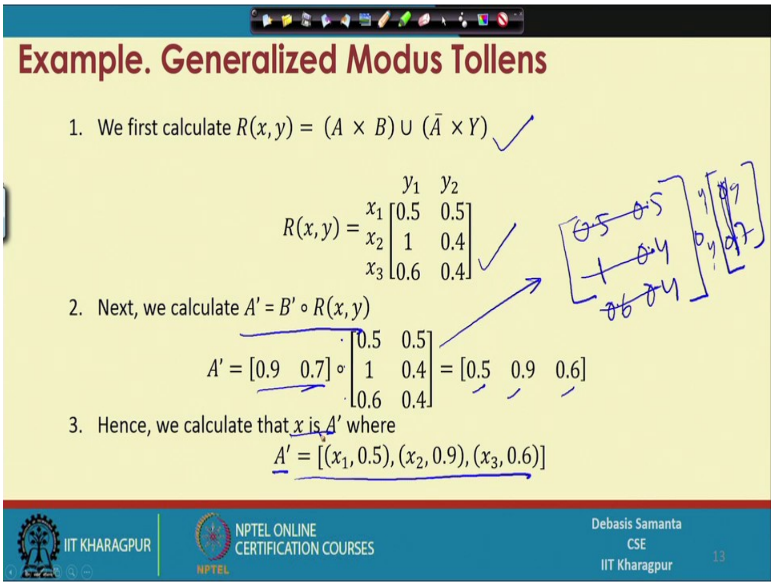
(Refer Slide Time: 12:37)



Now, again the same example I can use to illustrate the GMT. So, these are the two universe of discourse X and Y. And this is the rule giving the relation between x is A and y is B and A and B are the two fuzzy sets it is given like this.

And you have given y is B' where the B' takes like this form and we have to compute the proposition $x \text{ is } A'$. So, here basically B' is given. So, compute A'. So, you have to apply G M.

(Refer Slide Time: 13:15)

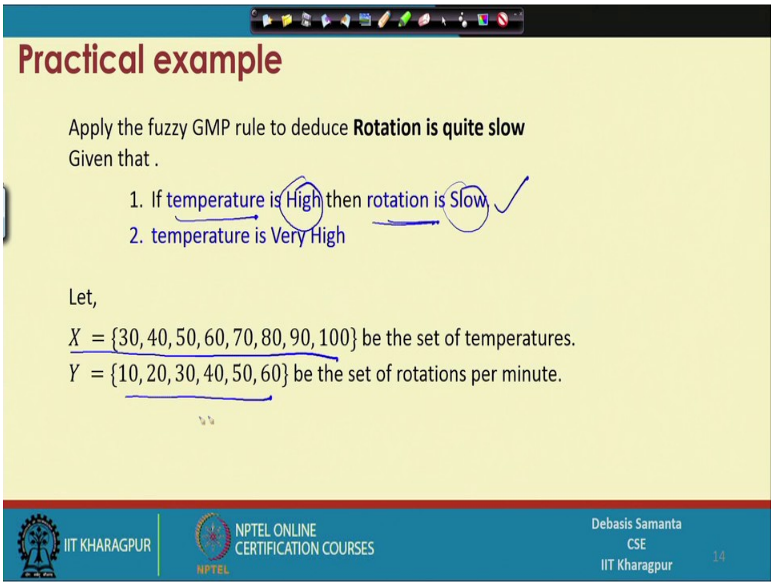


So, this is basically takes this form. So, this relation matrix can be calculated which can be obtained like this and then applying GMT we have to calculate this one. That means, $A' = B' \circ R(x,y)$. Now here now here you can check it this basically not applicable because it is the number of elements 2 and number of elements 3.

So, actually we can alternatively we can write it because A' should have the elements x_1, x_2, x_3 here x_1, x_2 and x_3 is there. So, I can write it basically this a 0.5, 0.5, 1, 0.4 and 0.6 and 0.4, then the composition and this composition I can write either this way or we also we can write in this way where this is basically y_1 and y_2. So, it is 0.9 and 0.7 then the different element can be obtained applying this one and this one in direction; direction using max-min composition. So, if we apply this one this one then the first element 0.5 can be obtained if we apply this one and this one the 0.9 can be obtained and this one this one 0.6 can be obtained.

So, this way we shall be able to obtain A' given $R(x,y)$ and B' and which takes this form. So, this is basically x is A ' another proposition. So, you can infer from two rules or propositions another proposition the similar idea can be extended to two rules also we will see it shortly. Now so we have understood about the GMP and the GMT the generalized modus ponen and general modus tollen are the two tools.

(Refer Slide Time: 15:08)

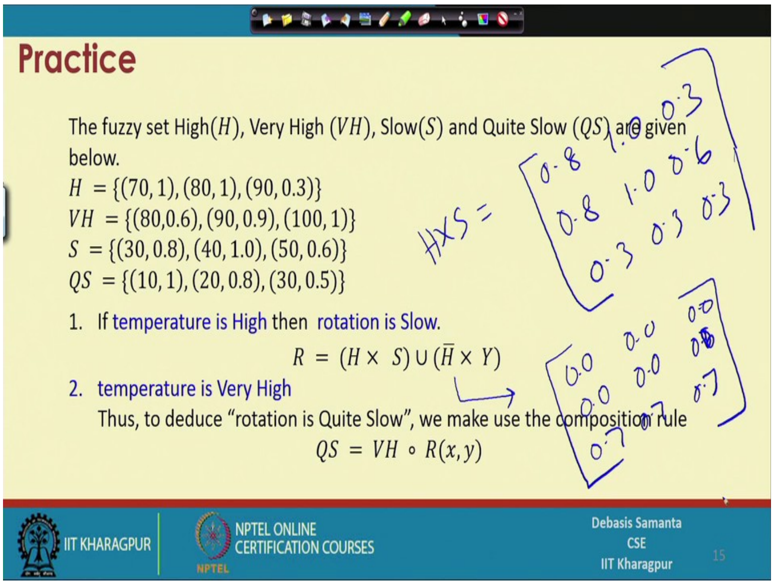


For the calculation of or for the inferences of some other rules. Now here, I just want to conclude with another example and let us see. So, this is an example here we have considered.

Say suppose these are the two rules are given there and this is the one example it is there given the two rules here if temperature is high and rotation is slow. Now here this high right is a basically one fuzzy sets and slow is a one fuzzy sets, this high and slow the two fuzzy sets are defined over two discourse one is the discourse of temperature, another discourse regarding the rotation. So, here basically X is the universe of discourse regarding the different temperature there is basically sets and then the rotation rotation has certain metrics. So, they are expressing this form 10, 20, 30 like. So, X and Y are the two universe of discourses representing the temperature and the pressure respectively.

Now, if I want to express the high temperature then definitely we have to define one fuzzy sets high temperature. Similarly, if we define the another fuzzy set, say rotation slow or slow rotation we can define this one. Likewise, High there may be another fuzzy set can be defined Very High provided that their difference in the sense that the different membership values, for the different elements. Now let us see; what are the different fuzzy sets that we can conclude here for example, here.

(Refer Slide Time: 17:00)



So, suppose temperature with the universe of discourse as we have discussed 10 to 100 and we define the two fuzzy sets regarding the high temperature and very high temperature.

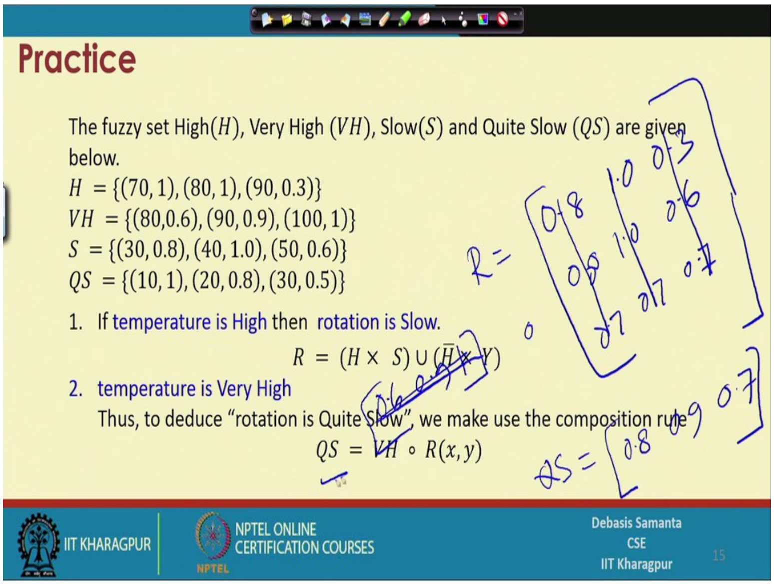
So, this is basically one fuzzy set high temperature and this is the another it is called the very high temperature and you can see that difference is there it is like either the in terms of elements and then degree of membership or both like. Now likewise regarding the rotation as the universe of discourse we define two fuzzy sets one is called the Slow and another is Quite Slow. The slow fuzzy set is defined using this form and then quite slow is defined this form. Now, so the rule temperature is high then rotation is slow it can be expressed using this expression relation that is the according to Zadeh's max-min composition relation. So, R this is basically the relation matrix R showing if temperature is high then rotation is slow. We can calculate these value.

And then if temperature is very high is given then we have to conclude some other in the universe of discourse slow. So, it is rotation is quite slow. So, that means, this is the one premise is given and this is another premise is given and we have to derived another premises or proposition rotation is quite slow. So, we can apply GMP in this case because this is this is to given and we have to derive this one x is A, y is B, x is A' you have to obtain y is B'. So, we can apply GMP and this is the final formula that can be used to calculate this one. Now if we take this calculation, we can check that the result that can be obtained we can check that.

So, here basically $(H \times S) \cup (\acute{H} \times Y)$. So, $H \times S$ it can be calculated if you can check it I am giving the final result. So, I advise you to check yourself. So, this is 0.8, 1.0, 0.3. It can be calculated 0.8, 1.0, and 0.6 and 0.3, 0.3, and 0.3. So, this is basically gives the calculation $H \times S$. Now, similarly this also can be calculated and we can obtain another relation matrix that can be obtained as 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, and 0.7, 0.7, 0.7.

You can verify this calculation now once you know this one then we can take the mini the union operation that means taking the value.

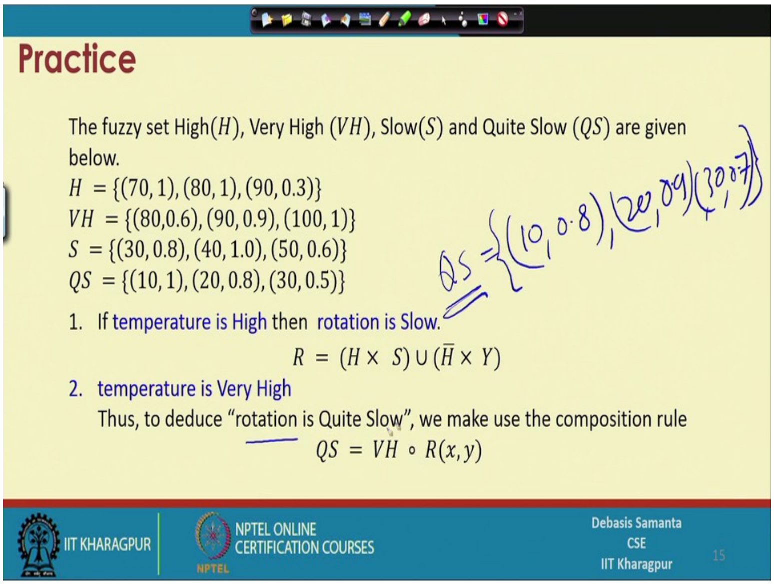
(Refer Slide Time: 20:13)



So, this relation matrix R, then can be expressed R this is equals to another relation matrix you can verify this relation matrix that can be given a 0.8, 1.0, 0.3, 0.8, 1.0, 0.6, 0.7, 0.7, 0.7.

So, it is given then it is given like this one now we have to. So, this R is available now. So, we have to obtain the QS. So, we can use this formula $VH \circ R(x, y)$. So, basically composition VH is basically this one. So, you can write 0.6, 0.9 and then 1. So, again the Max-Min rule can be applied and then QS this can be obtained this and this it will give the first element you can say 0.8 then this and the we can obtain 0.9 and this and this you can obtain 0.7.

(Refer Slide Time: 21:29)



So, this finally, QS can be calculated. So, QS that can be obtained. Therefore, it is a universe of discourse this one. So, we can write (10, 0.8) then (20, 0.9), (30, 0.7). So, this is basically the fuzzy set expressing the QS or rotation is quite slow giving the fuzzy set it is like this.

So, this is the way that we can use the GMP, and GMT to infer fuzzy relation. Now, so far we have discussed about the fuzzy implication or inferences; rather inferences given one rule and one proposition. In the next lecture, we will discuss about the fuzzy implication whenever the two or more rules is given, and then how another rules can be obtained from them.

Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

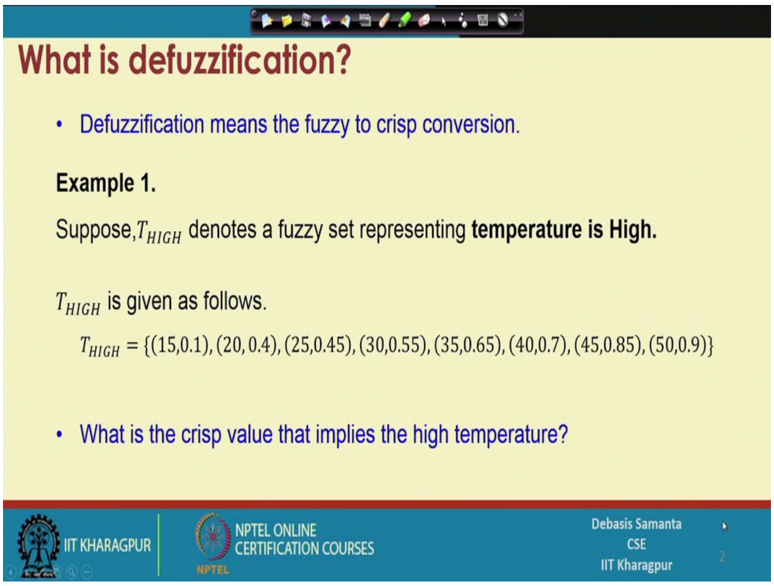
Lecture – 09

Defuzzification techniques (Part-I)

Now, we have discussed about the different elements that can be derived. So, either these elements is a fuzzy set or is a fuzzy rule or is a relation or proposition or some inferences whatever it is there. Now whatever the elements that is there, they expressed in the form of a fuzzy concept right, but the idea is that our we can understand instead of fuzzy rather the crisps, whenever we have to use this fuzzy value then they should be converted into the crisps value. So, in today actually we will discuss about how a fuzzy value can be converted to the crisps value that is this discussion basically involved a lot of technique to be discussed. So, we will take maybe two lectures two I mean to discuss the whole topics.

So, first we can consider the first part of the discussion. So, we say the discussion defuzzification techniques part 1.

(Refer Slide Time: 01:20)



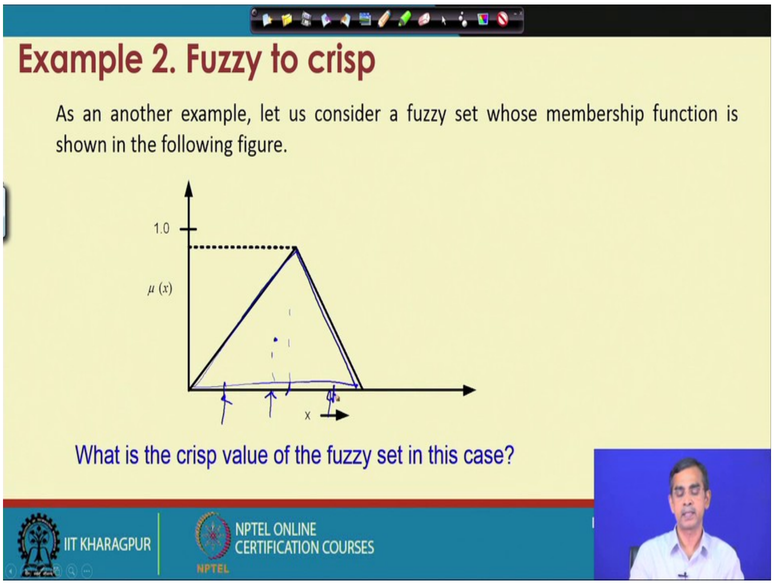
And So, let see how it is basically the concept of defuzzification it is actually. As I told you defuzzification means from a fuzzy value to a crisps value.

Now, as an example I can say T_{HIGH} is a fuzzy set the temperature is high representing and these fuzzy sets take these form. So, our objective is that, if this is the fuzzy set then exactly what is the high temperature in the concept of crisps value? That means, what is the crisps value that implies high temperature?

So, one way if I ask you, you can give that 50.09 is the high temperature because it has the highest what is called the high value of the membership degree of membership. But some people can say that I can take neither very low value nor high value or there are medium value; that means, this one; that means, this one. So, let me two value and then take the minimum or average of the two value. So, this one whatever it is there or a middle value one also can be taken as the high temperature.

Now, these are the basically guessing and in let see in fuzzy theory, how such a crisp value can be calculated rather with more mathematical tuning.

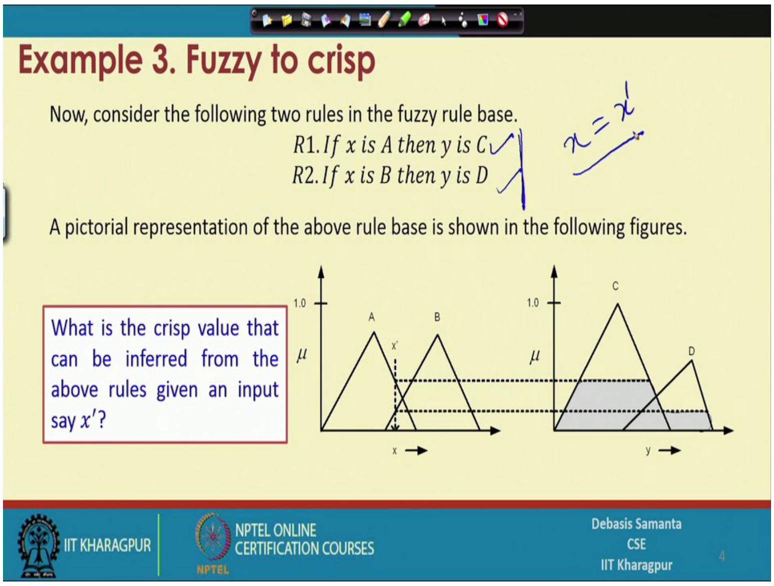
(Refer Slide Time: 02:49)



Another example, say suppose this is the one fuzzy set is given in the form of a graph. So, these basically describe a fuzzy set what with these continuous values of element x, then what is the crisps value. That means, whether crisps value this one or this one or this one or it is somewhere this one. So, crisps value is this one or this one or this one which one it is basically.

So, let us see how the fuzzy mathematics or fuzzy logic provides as a way to calculate the crisps value for such element like a fuzzy set shown in the form of a graph.

(Refer Slide Time: 03:33)



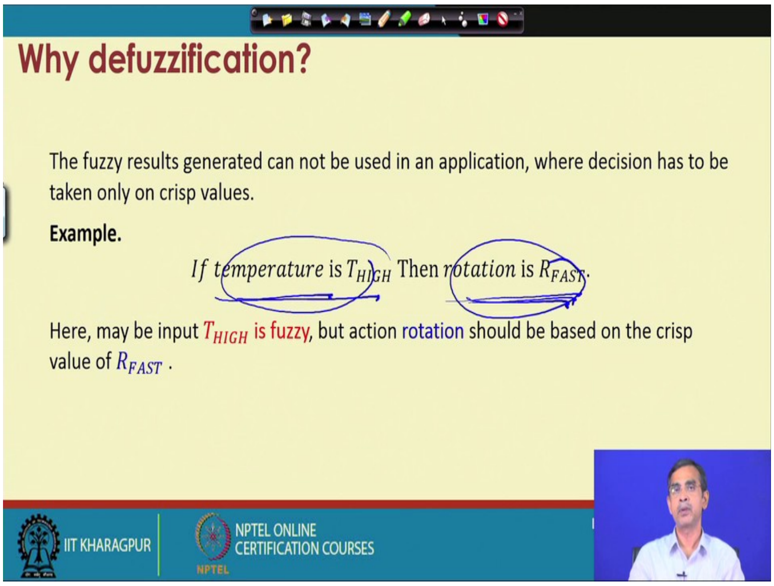
Now another example, here also we have to obtain the crisps value here these example is more complex of course. So, idea is that these are the two rules is given these are rule right then what is the output. If **x is A then y is C** and **if x is B then y is D**, then if these two rules are given then for x is equals to say some element x'; that means, that $x' \text{ is } A$ **then y is C** and x' **is B then y is D**. then what is the final crisps conclusion from the two rules that is fireable to $x = x'$.

So, this rule can be computed again using the defuzzification techniques, with the broad idea of the defuzzification technique is shown here. So, suppose this is the fuzzy set A defined over the same universe of discourse x and this is the B defined over the universe of this course x. And y is C; that means, C fuzzy set D fuzzy sets are defined by this and this one and they are defined over the another universe of discourse y.

Now, for some element x' right, this is the x'. So, we want to calculate the rule strength actually this is called the rule strength or the rule value. So, here basically if we see x' is fireable with this membership value $\mu_A(x')$ here and a $\mu_B(x')$ is here. So, if we draw a line and then this one. So, this basically **if x is A then, y is C** basically have this train up to this one. So, this one if x' is someone here then it will go there or above there, it will go there or it is here there.

So, depending on the values x'. So, the portion that will cover it is there. Now these portion, basically tell that what is the value of this rule? Likewise, if x' **is B and then y is D**. So, if we draw a line then we intersect this portion. So, these basically rules strength for this. Now taking the taking the merging of the two, so that two rules have the value this shaded area. So, this is the shaded area basically is the fuzzy value given this two rule. Now given this is the fuzzy value when how we can obtain the crisp value. So, this is the task and we can solve this kind of problem using the concept of defuzzification technique.

(Refer Slide Time: 06:30)

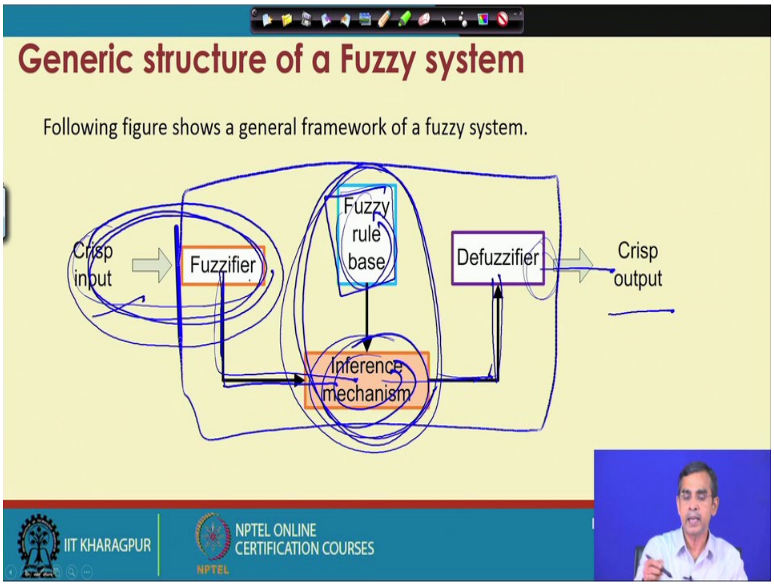


Now, why defuzzification or really it is very important or not that is also important. So, as I told you defuzzification require, because some application where it can take if the conclusion is available in the form of a crisps value. For example, if you want to develop one what is call the ac controller air conditioner call the controller. So, that controller basically controls depending on the temperature. So, the controller is design like this if temperature is high, then rotation is fast.

Now, temperature is high for a particular temperature is 40 degree then rotation is fast; that means, we have go for fast rotation. Now these value should pass to the controller in such way that controller can understand a particular value only that is this is the rotation say 30 rotations per seconds. So, then it will calculate. So, that if this is fuzzy and this is fuzzy then finally, we should have what exactly the defuzzified value or crisps value for this one. So, it is the usual this is the one example that I have placed here so that if something is fuzzily available, input or output whatever it is there finally, we need the fuzzy finally, we need the crisp value to be used for some final application.

Now, so this is basically the model that is followed in fuzzy system design as I told you here.

(Refer Slide Time: 07:59)



So, basically this is the fuzzy system that we want to design. Now fuzzy system takes the crisps input and then crisps output because we can think only in the terms of crisps value, and also we can conclude something if the crisp value is known to us. So, we are habituate with the crisp value rather than fuzzy value, but the fuzzy system will take only the fuzzy values.

So, the first step is that the crisp value should be converted to the fuzzy value. So, this is a fuzzified input, then from the fuzzified input it will go to the inference mechanism so; that means, if this is the input and they are the fuzzy rule base; that means, a set of rules we have already discussed about how the rules can be store in the form of a matrix. So, it is basically set of matrix given there. Now there is a technique regarding the inference mechanism we will learn about it later on and if the fuzzy input is given to this inference mechanism inference mechanism will discuss about the fuzzy rule base, then we will obtain another fuzzy output.

Now, this fuzzy output needs to be defuzzified and then the defuzzified result can be passed through the outside of the fuzzy system as a crisps output. So, basically is a fuzzifier and defuzzifier and then fuzzy rule base and inference mechanism, these are the four basic components or four basic task that is involved in order to developed a fuzzy system. Now the fuzzifier as you know, the fuzzifier is basically is a task of the fuzzy designer fuzzy engineer, who basically can convert crisp input to fuzzy input. So, it is basically in the form of a fuzzy system development, fuzzy engineer has to given idea how a crisp input can be stored in the fuzzy form.

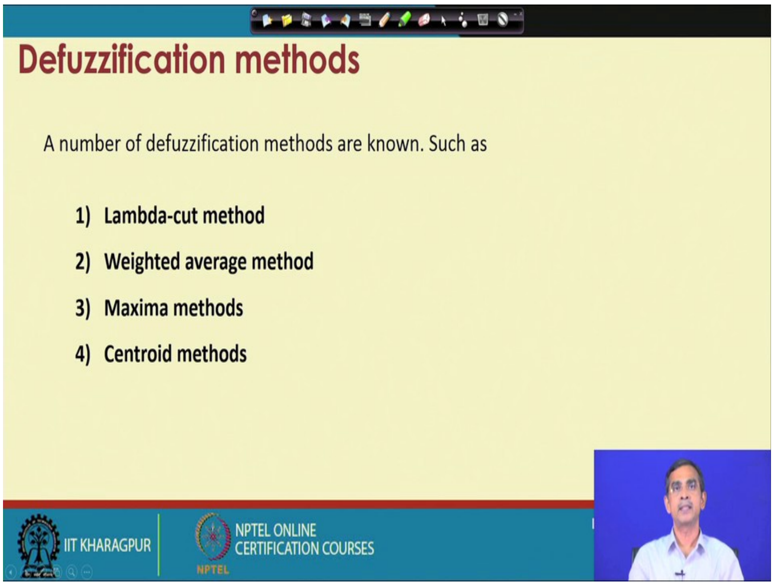
And then these are the basically fuzzy rule base system and then the inference mechanism can results the defuzzy. These ate the basically different fuzzy operations related the fuzzy sets related to the fuzzy rules, fuzzy proposition and then fuzzy implications. Finally, the different inferences can be obtained and these inference from the inferences. We will be able to defuzzified it so the crisp output. So, now, we are discussing basically how the fuzzy elements whether it is in the form of set or it is in the form of a relation matrix can be defuzzified, that the crisp result can be obtained.

So, let us see what are the different techniques are available so far the defuzzification techniques concern.

(Refer Slide Time: 10:43)

So, far the defuzzification techniques is concerned many methods are known we can broadly categorized the different methods, we have categorized four the first method is called Lambda-cut method.

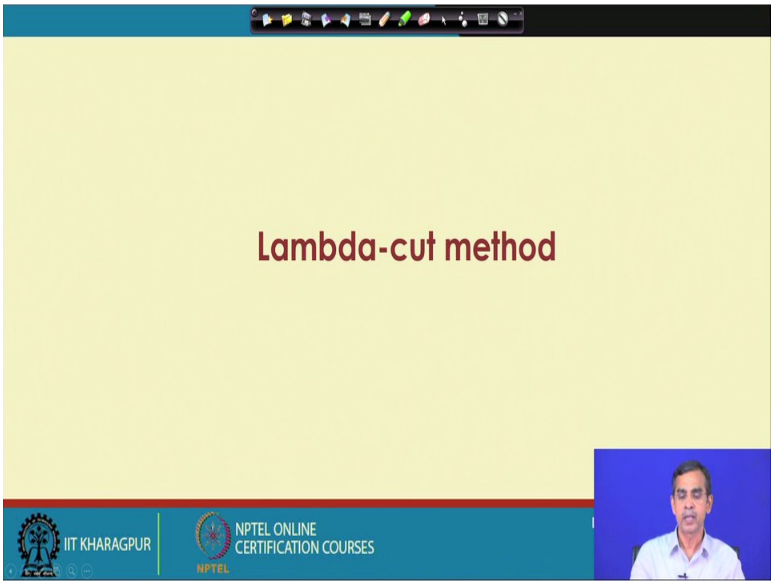
(Refer Slide Time: 10:49)



The second method is Weighted average method, and the next is Maxima method and finally, is the Centroid methods.

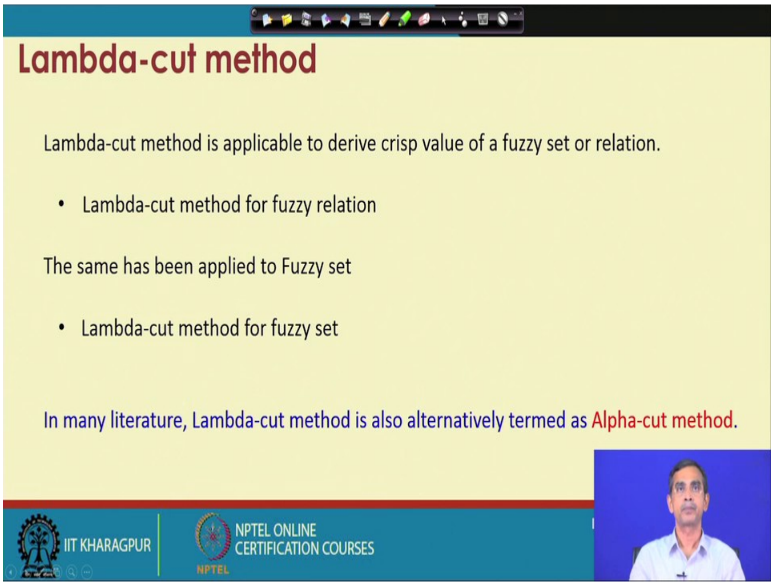
Now, these are the different methods the way a fuzzy value can be converted to its corresponding to its crisp value. So, different methods have their own merits as well as our limitation, we learn about it whenever we discuss the different methods one by one.

(Refer Slide Time: 11:25)



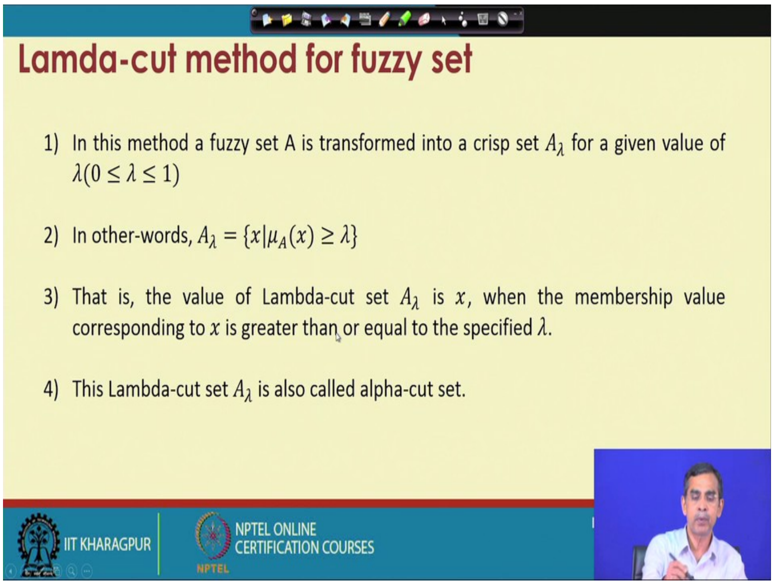
Now first we will discuss about the Lambda-cut method

(Refer Slide Time: 11:29)



Now, Lambda-cut method is a very popular method already known in the context of crisp theory crisp algebra; that means, in the context of set algebra or Boolean algebra. The same thing can be extended and the context of fuzzy theory. So, Lambda-cut method for fuzzy relation and Lambda-cut method for fuzzy set.

(Refer Slide Time: 11:54)

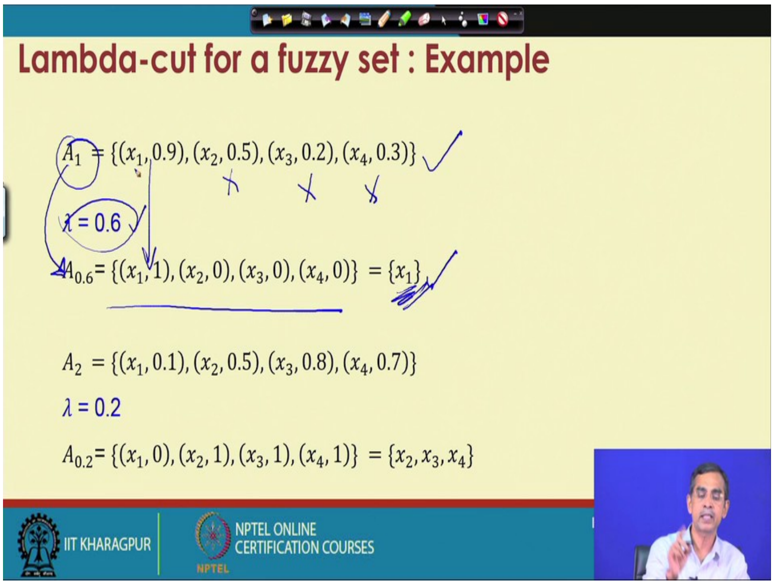


We will discuss about the Lambda-cut method for fuzzy set and then Lambda-cut method for fuzzy relation.

Now, let us first discuss about the Lambda-cut method for fuzzy sets so that from a fuzzy set to crisp set can be obtained. So, idea it is. So, in this method the fuzzy designer should choose one value, the value is called lambda that is why the method is called Lambda-cut method. So, lambda value should be in the range 0 to 1 now. So, then the Lambda-cut method it is basically we have discussed when we discussed about the fuzzy terminology, that is basically A_λ is A_α cut like is. So, it is A_α actually α is λ here. So, A_λ and A_λ is basically a crisp set which can be obtained that it includes all the elements $\{x \vee \mu_A(x) \geq \lambda\}$. That means, from the given fuzzy set we can find a crisp set so that the elements whose membership value is $\geq \lambda$.

So, this is the value of lambda set and. So, this is also depend on the A_λ . For the different values of λ definitely A_λ will be different. Now, so, A_λ is also alternatively called alpha-cut set, because we have discuss about the cut set of a fuzzy set. So, it is a alpha-cut set. Now let us see one example so that we can discuss about this Lambda-cut method.

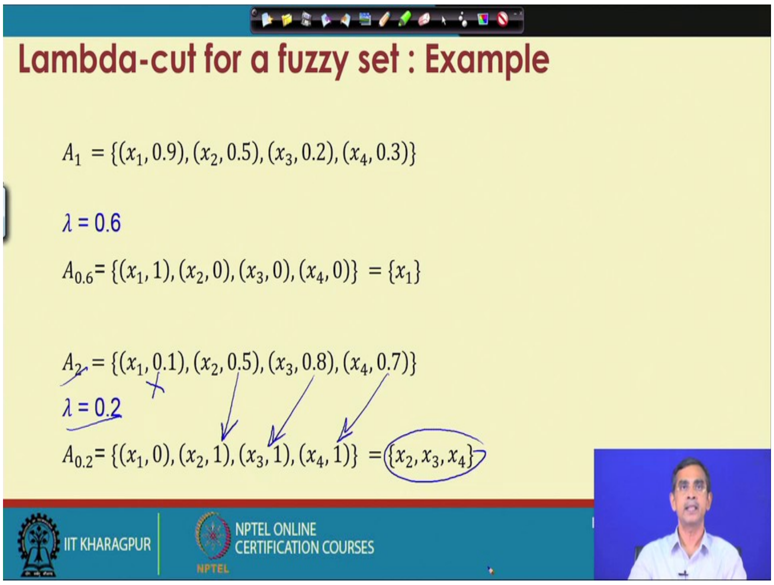
(Refer Slide Time: 13:36)



Yeah. So, these example say. So, A_1 is a fuzzy set and let us consider λ is 0.6. So, it is A_λ that is the lambda cut according to the value of λ these one. So, it will basically take all the elements, which whose membership value is greater than or equals to 0.6.

So, this way x_1 is qualified. This is not qualified. This is not qualified. This is not qualified. This is not qualified. So, we can convert these fuzzy sets into the this is the crisp set or more general form the crisp set can be express these one, because those are the 0 they should not be there and those are the element one these one. So, A_1 if this is the fuzzy set and then if we follow certain value of λ , then the crisp set can be obtain and this is the crisp set. That means, if say the temperature is high these one, then the high temperature will be x_1 temperature like this one. So, it is the example like this.

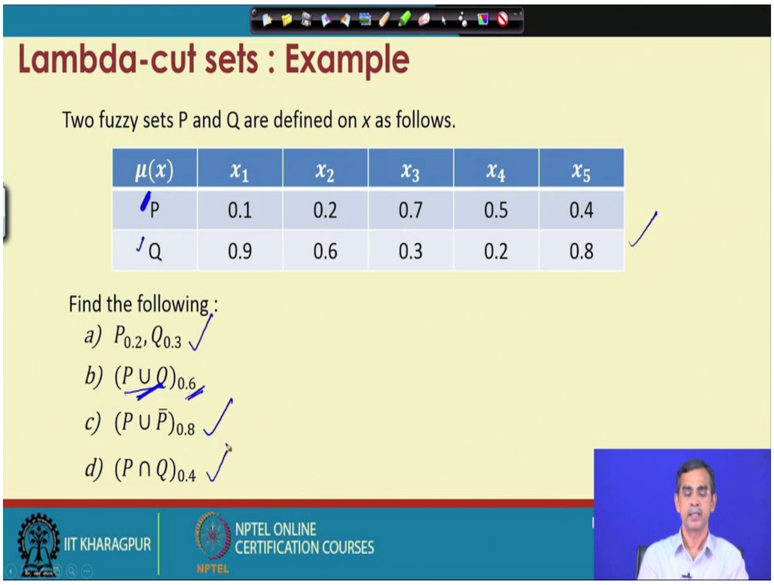
(Refer Slide Time: 14:38)



Now, as a second example. So, suppose this is another fuzzy set A_2 and we take λ it also 0.2. That means, you will take those elements whose membership value is greater than or equals to 0.2. So, it is not qualified. This is qualified. This is also qualified and this is also qualified. These means that the A_2 given the fuzzy sets and λ which is specified a 0.2 the crisp set that can be obtained this is the crisp set. Now you can say that. So, lambda cut gives you a crisp set, crisp set can be a NULL or it can have one or more elements in the set.

Now, in case of one or more element, they are basically all the equivalence are as the in the in the context of crisp that if this is the fuzzy then this is the crisp value either it is $x_2 \vee x_3 \vee x_4$ or you can take the mean of the three values to conclude precisely that this is the crisp result. So, it is the idea about the Lambda-cut method and this is the in the context of fuzzy sets.

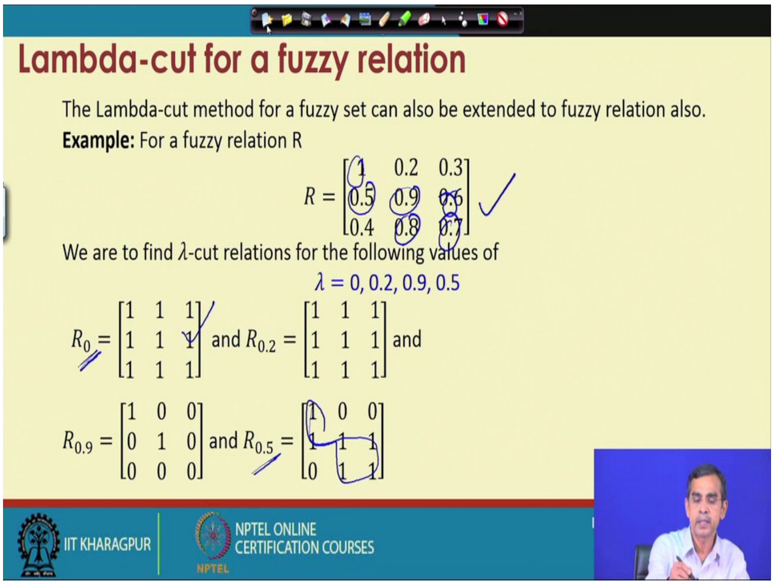
(Refer Slide Time: 15:45)



Now, the same idea can be calculated for the other complex fuzzy forms, here as an example. So, suppose you have given two fuzzy sets P and Q with the membership value it is shown here in this table. Then definitely will be able to calculate these things without previous knowledge.

Now, if we have to calculate $(P \cup Q)_{0.6}$. So, the idea is that we first obtain the fuzzy union of these to fuzzy sets P Q and then for the resultant fuzzy sets we can apply 0.6 then we can obtained the result of this $P \cup Q$ for this 0.6 lambda set similarly these one and these one also can be calculated . So, idea can be extended to the complex formulation relating to two or more fuzzy sets as well.

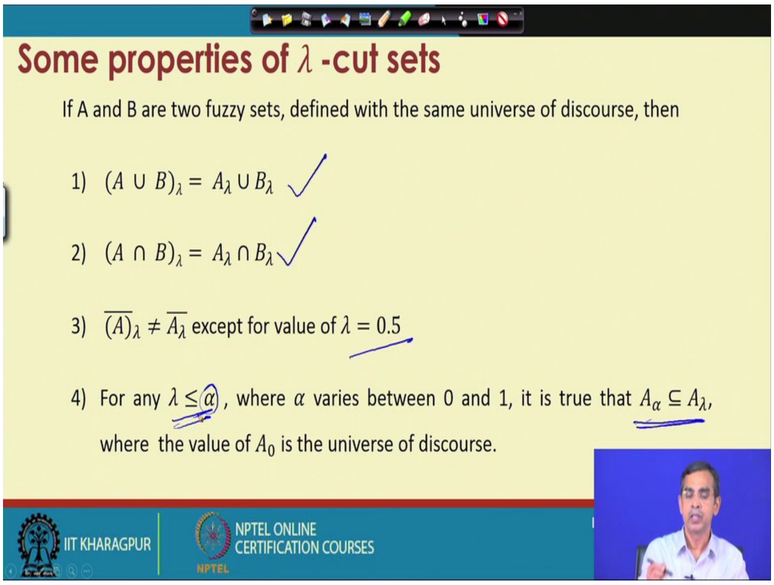
(Refer Slide Time: 16:40)



Now, we can again extend the same concept of Lambda-cut set to fuzzy relation or in a relation matrix. So, idea it is like this. A relation matrix we know it is suppose for an example a relation matrix is given here it is take this form. Now again a relation matrix a if a the lambda-cut for a fuzzy relation can be specified with respect to a specific value of λ , I am giving here three examples four examples. So, if λ equals to 0 then how the crisp relation can be obtained. So, if λ equals to 0 that means, will take only those entries whose basically value greater than 0. So that means, in this case if. So, this is R_λ , λ equals to 0 and basically all entries are qualified. So, this is the crisp relation matrix.

Now, again say suppose 0.5 if we take; that means, we can take only those entries whose value is greater than 0.5. So, these one, these one, these one, these one, these one and these one. So, these ways these these these are the different values that is there in the crisp relation. So, this is a fuzzy relation and these are the different crisp relation. As you know the fuzzy relation can have entries in between 0 and 1 both inclusive. Whereas, crisp relation is basically have these one and so it is. So, is a crisp. So, is a fuzzy. So, fuzzy to crisp can be calculated using the lambda cut method they like this one.

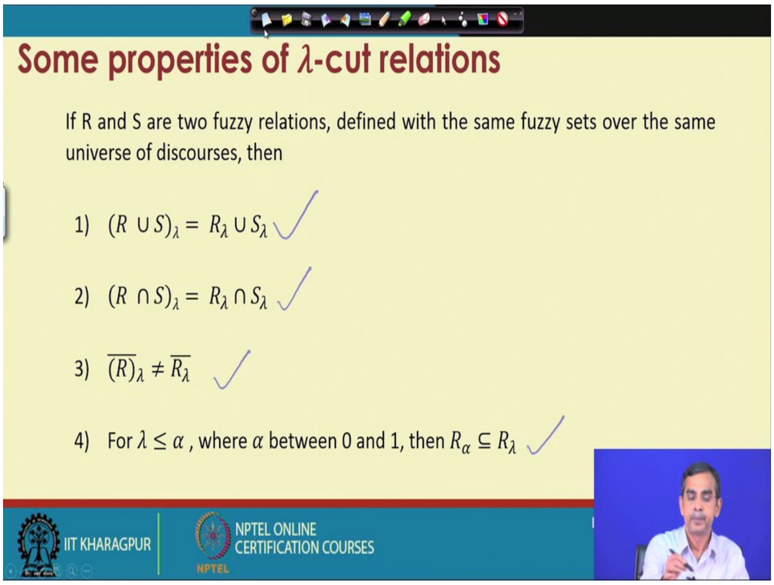
(Refer Slide Time: 18:22)



Now, the same thing can be applied to other things also there are some properties hold good. So, far the lambda cut set is their, these property needs to be understood right. So, basically this is the one property that if $(A \cup B)_\lambda$ is taken this is basically equivalent to take the $A_\lambda \cup B_\lambda$. So, this is basically one rule also applicable this rule also just it for union it is for intersection and one thing is that $(\acute{A})_\lambda$, and then $\acute{A}_\lambda$ they are not generally equal, but they are only equal for $\lambda = 0.5$. And for any $\lambda \leq \alpha$, α is another value in between 0 and 1 we can check we can prove that $A_\alpha \subseteq A_\lambda$ if this in equality holds.

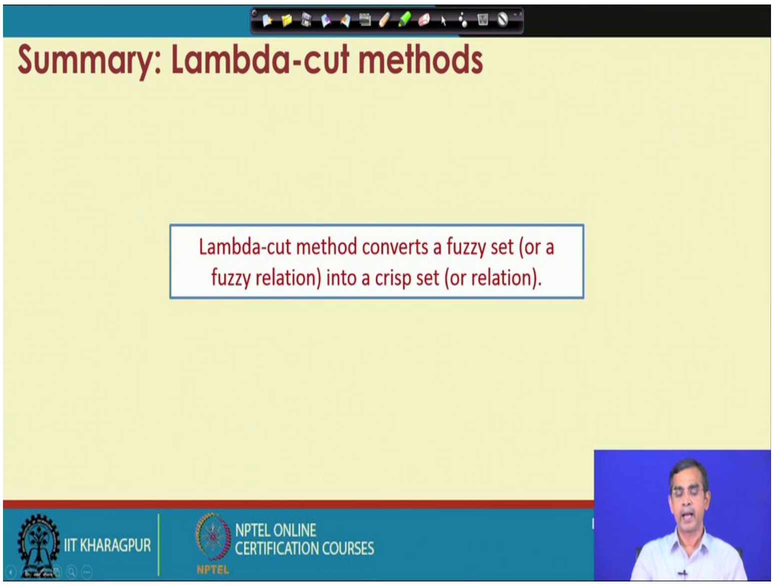
So, these are the properties it basically holds good, for the lambda cut technique is concerned.

(Refer Slide Time: 19:25)



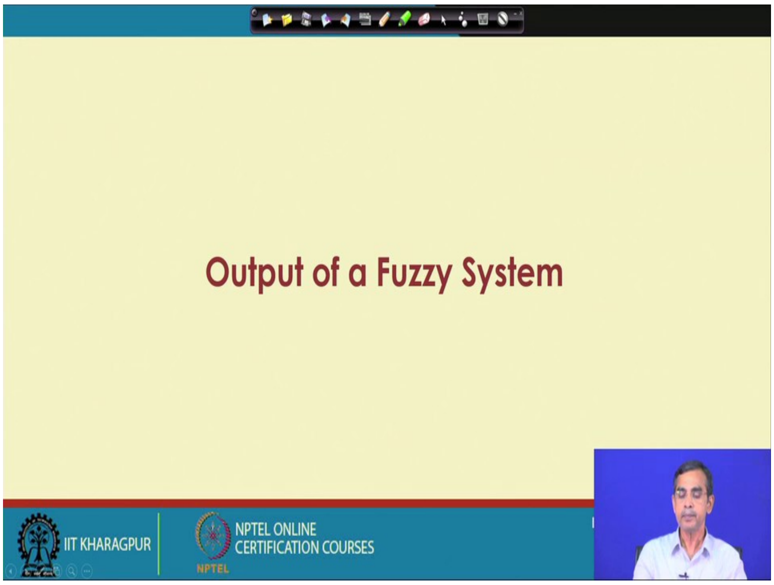
Similarly, for the relation these are the properties also satisfied, this property is for union and this is for intersection and this is for the compliment and this is the subset of this relation.

(Refer Slide Time: 19:43)



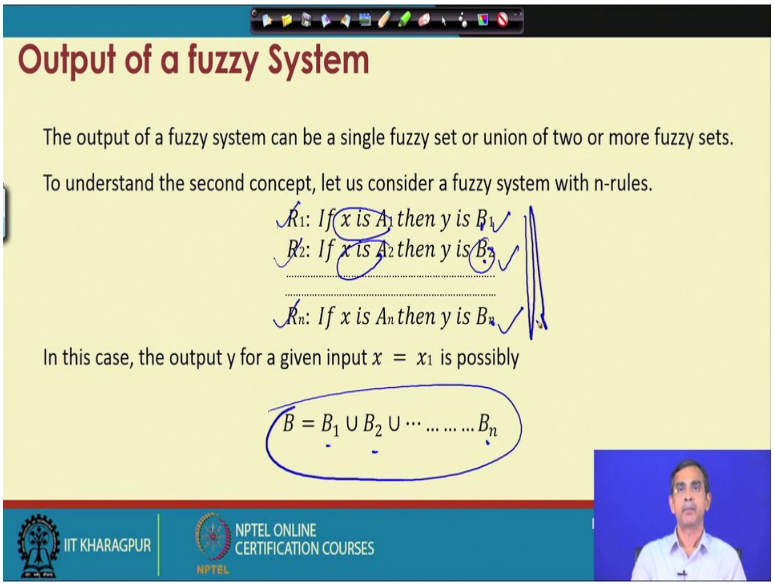
So, this is the Lambda-cut method in general and we can conclude one thing regarding the Lambda-cut method is that, Lambda-cut method converse a fuzzy set for a or for a fuzzy relation into crisp set or crisp relation, basically fuzzy set to fuzzy crisp set or a fuzzy relation to crisp relation.

(Refer Slide Time: 20:02)



Now, before going to discuss the other technique, I will discuss about output of a fuzzy system. How basically the output of a fuzzy system can be concluded or can be calculated given many fuzzy elements or more precisely we will discuss about many fuzzy rules are there.

(Refer Slide Time: 20:25)

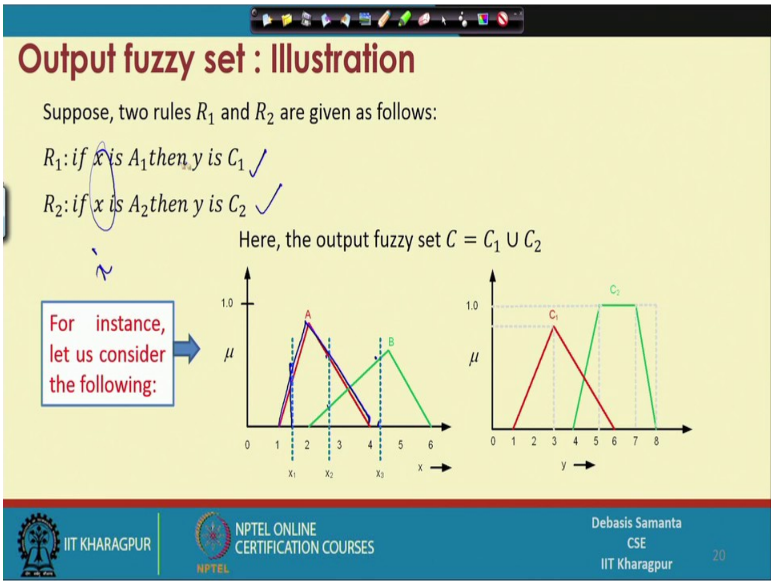


So, if more rules are given then how you can conclude the what is the result is there. That means, for a given input say suppose the n rules are applied. So, for a input R_1 is also satisfied R_2 is also true, R_n is also true. So, n number of rules are true for a particular input then what is the output.

So, output is basically. So, if it is a rules satisfied giving B_1 , B_2 , B_n then output can be like this that it is basically union of B_1 , B_2 , B_n but not exactly the union it is basically the rules strength related to x is A_1 that B_1 . So, it is not the entire B_1 , but a part of the B_1. x is A_2 not the entire B_2 , but a part of the B_2 . So, basically fuzzily it is part of B_1 part of B_2 part of B_n . So, is a part of B_1 part of B_2 part of B_n and if we take the union of all these parts then finally, it will gives the total what is call the parts that all these rules basically signify for the output.

Now, here is the idea about that how these kind of calculation is possible and then they are corresponding a crisp value.

(Refer Slide Time: 21:42)

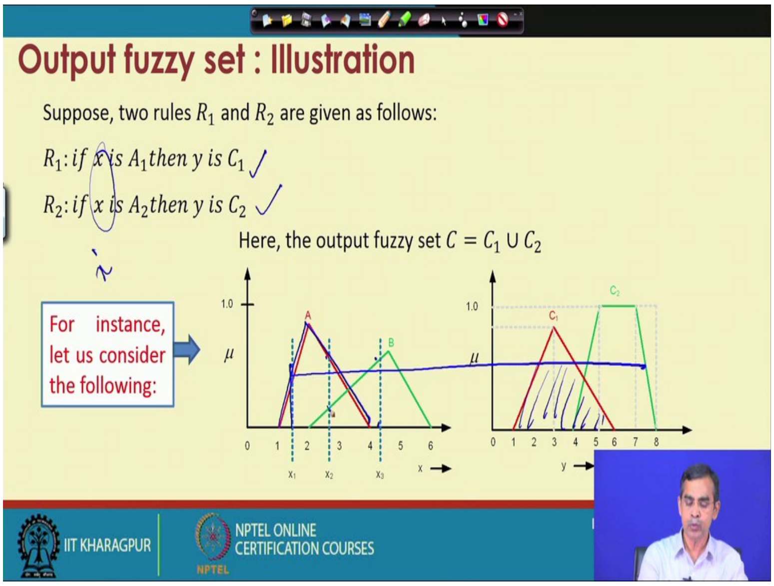


So, this idea can be mathematically better described, and I just discuss in terms of two rules the same idea can be explained more than two rules as well. Now here is an example we can take it clearly. So, suppose two rules are given here the two rules is $if\ x\ is\ A_1\ then\ y\ is\ C_1$ and another rule is $if\ x\ is\ A_2\ then\ y\ is\ C_2$.

Now, for an input say x equals to some x '. We have to calculate what is the rule strength. Now here you can see I just. So, graphically I can calculate it or graphically I can display it. So, it is basically the fuzzy set A and this is the fuzzy set B and here the C_1 and C_2 other two fuzzy sets C_1 and C_2, suppose they are defined over the same discourse x and this the rule C_1 and C_2 fuzzy set defined over the discourse y and these are the graphical representation of their sets now. So, if x equal to x_1; that means, it qualify this one. If x equals to x_2 then it fire both A and B and if x equals to x_3 it fires only set B, but not set A.

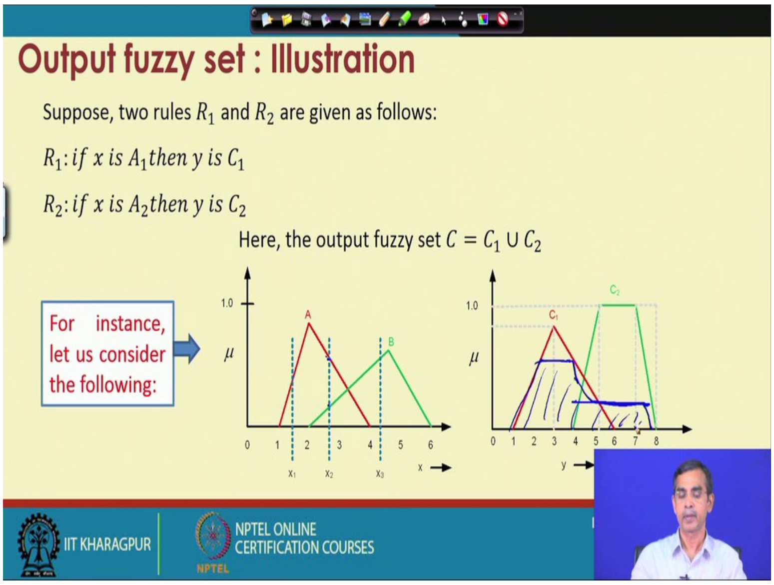
Now, so, the three different situations I have mentioned here if for a given value only **if x is A then y is C** it is fired, but this B is not fired only one. So, in that case if we draw a line parallel to these.

(Refer Slide Time: 23:20)



So, as it is only. So, related to C_1. So, this is the part of the set C_1 that is basically the rule strength of this one. Now again if x_2, if x is C_2 when it satisfied both part.

(Refer Slide Time: 23:41)

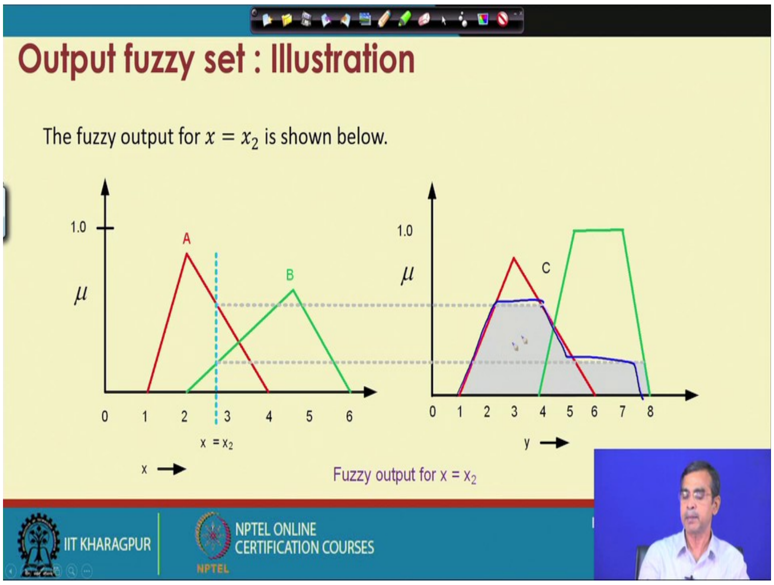


So, if we consider this part then this is the C and if we take this part when it is this one.

So, the rules strength is basically the total part is these one. So, this is the basically the resultant fuzzy set. Now, the next is x_3 is basically only these on a does not satisfy these means these rule does not fire, but these rule fire in this case, and then we can see this part then it is basically these part. So, what this rule the rule strength will be these one; that means, this is the fuzzy output.

So, we have some idea about that for the given rules rule or set of rules, how the fuzzy output will be there. So, geometrically a fuzzy output is basically is a geometrical shape and these basically a portion like this portion or is a curve like. So, this is also a fuzzy set I mean like how look like this. So, you have to obtain the crisp value from this fuzzy sets like. Now let us see how such a things can be extended in more general way, I can the extend the same idea around again here.

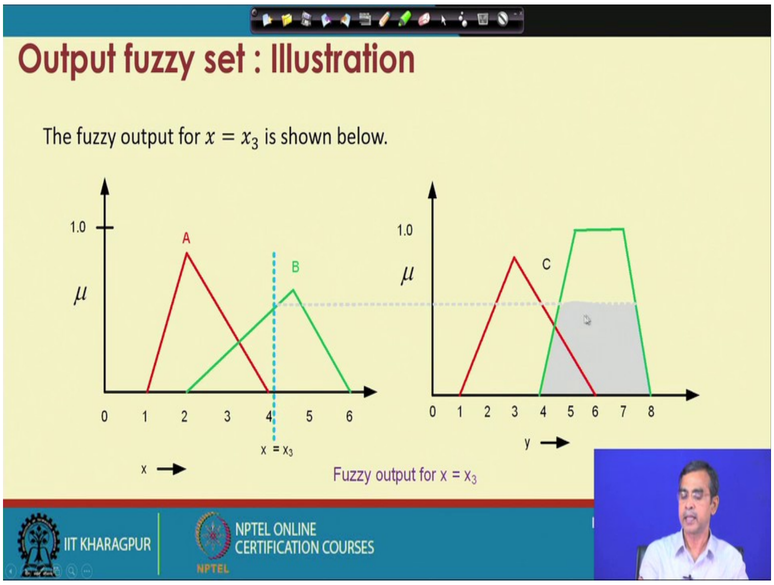
(Refer Slide Time: 24:52)



So, as I told you if x equals to x_1. So, this is the only part that is covered; that means, **if x is A then y is C** the fuzzy output is this one is the fuzzy output in this sense.

Now, if the as is another example if x is x_2 then the fuzzy output will be these one. So, this is the fuzzy description of the output and the second case is if x equals to x_3.

(Refer Slide Time: 25:30)



Then this is the fuzzy output. So, this is the fuzzy output fine. So, we have discussed about the fuzzy defuzzification technique, and particularly we discuss in this topics the Lambda-cut method and then how the output of a fuzzy system can be calculated. We will discuss the second part of the defuzzification technique particularly if a different geometrical shape is given to us a which represents a fuzzy output, how the crisp value can be calculated that will be discuss from our next lecture.

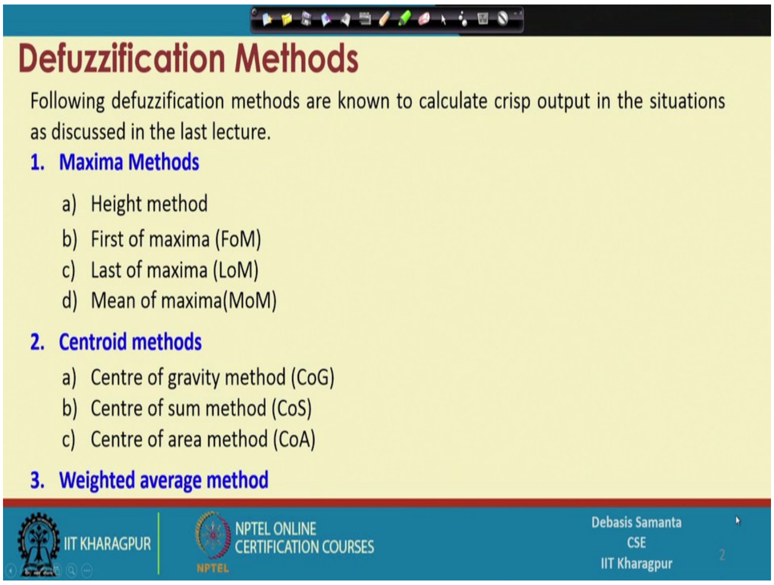
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 10
Defuzzification techniques (Part - 1) (Contd.)

We are discussing about defuzzification technique. Defuzzification technique is required in order to find a crisp value for whether it is a fuzzy set or it is a fuzzy relation or it is a fuzzy rule. So, in the last lecture we have studied Lambda-cut method to find a crisp value for a fuzzy set and fuzzy relation fuzzy rule. And also we have discussed about how a fuzzy rule can be graphically displayed and there are certain graphical way to find the output of the fuzzy rule. And then today we will discuss about if a fuzzy rule is portrayed in the form a graph, then how we can obtain the corresponding crisp value.

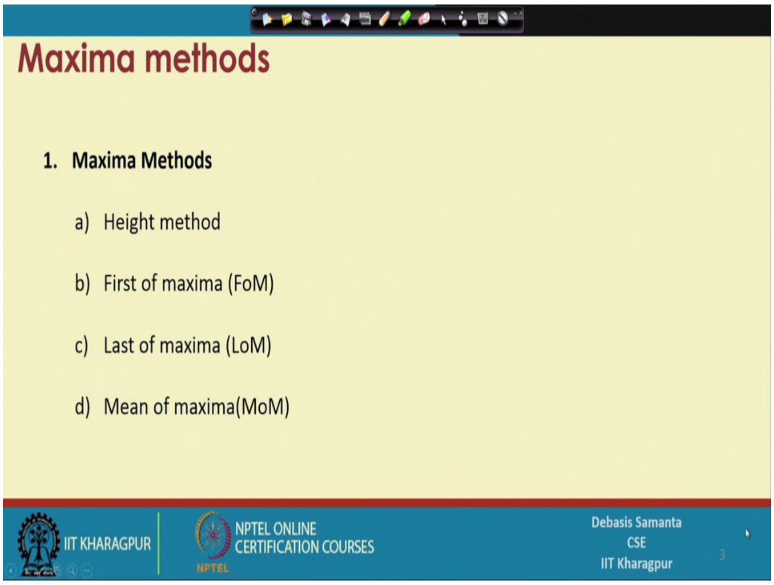
(Refer Slide Time: 01:10)



Regarding this fuzzifiaction defuzzification technique there are many methods other than the Lambda-cut method that we have discussed. So, all the method can be categorised in to three broad categories.

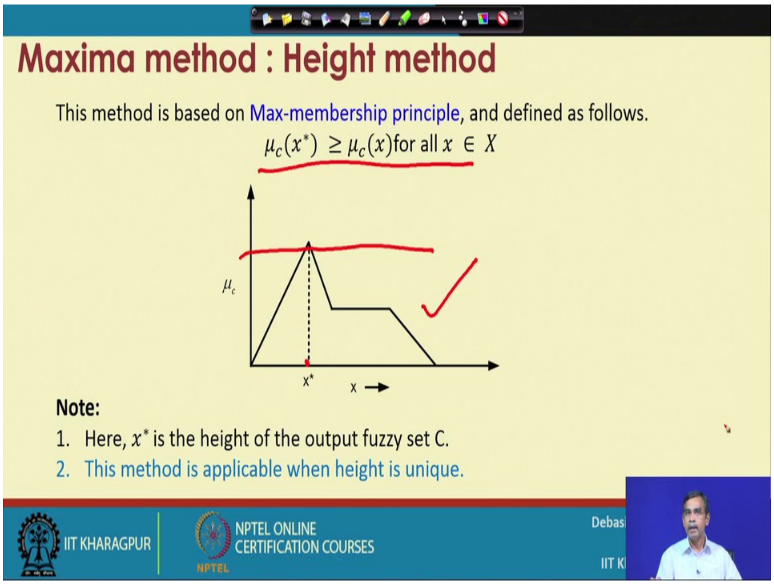
They are maxima methods, centroid methods and weighted average method. So, so far the maxima methods are concerned, there are again many methods like height method, first of maxima, last of maxima and mean of maxima method. Again so far the centroid method is concerned, there are three popular methods; centre of gravity method, centre of sum method and centre of area method weighted average method is only one approach. Now although we will discuss in the form of a graph, but actually the all methods belong to these categories can be obtained is in numerically also. So, you will get an idea about how numerically the methods can be applied, but we will initially learn the graphical approach and then I will give an idea about the how the numerically the same thing can be obtained.

(Refer Slide Time: 02:24)



So, let us first discuss about the maxima method. As I told you belong to this method there are four different approaches. One is height method, first of maxima method, last of maxima method and mean of maxima method.

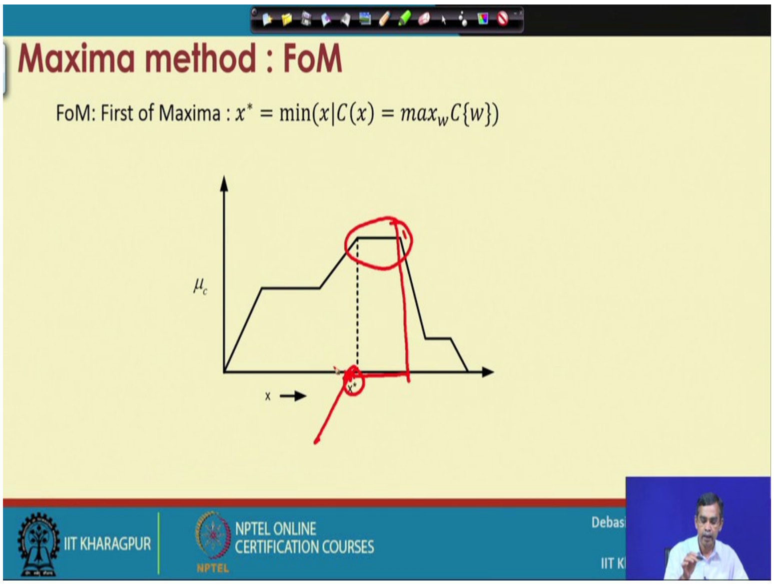
(Refer Slide Time: 02:42)



So, let us first discuss about the first method the height method. So, these slides show how the height method will look like this. This method in fact, based on the max membership principle. So, Max-membership principles can be expressed within this expression. So, it basically find what is the maximum values of membership for an element and the element which has the maximum value it becomes a crisp value for that. As an example, suppose this is the one graphical display of the fuzzy sets and if we see for the different elements. So, there is an element $x^{¿}$ for which the membership value is high.

So, this means the crisp value for this set will be $x^{¿}$. So, here in fact, you can see $x^{¿}$ having the highest membership value is become the height of the fuzzy set. So obviously, we can observe that this method is applicable only when a unique height is applicable. So, if there are more than one height. So, we have to follow some other method.

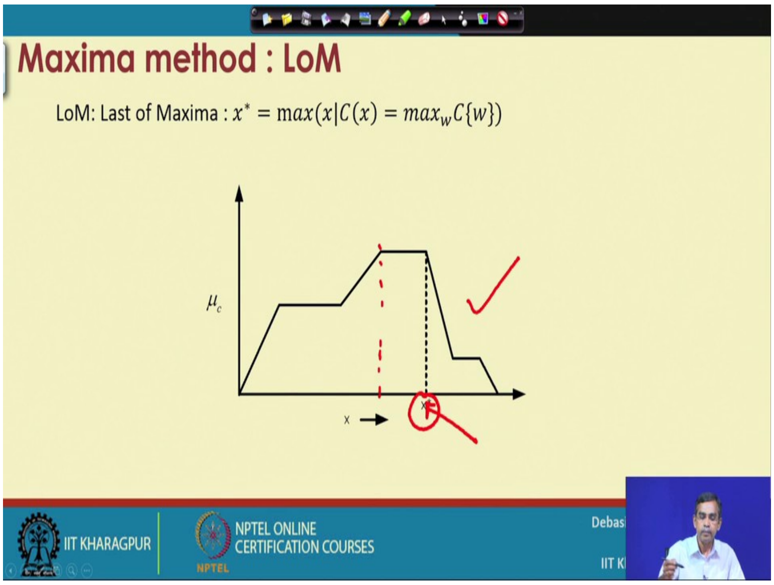
(Refer Slide Time: 04:12)



So, method is basically the first method in this line is called the first of maxima method. It is the method by which if we have more than one values having the highest membership values, then we have to take that element which first highest value. So, again if we draw this graph, you can see within this portion, it has the highest membership value is this one.

So, the first element this is the $x^{¿}$ which has the highest membership is become the crisp value. So, in this case the crisp value will be obtained here for this fuzzy set. So, this is the first maxima method and this method can be mathematically expressed this one. Now let us consider another method it is called the last of maxima, it is just opposite to the previous method. So, in this method the element this is the largest element which has the highest values of membership value.

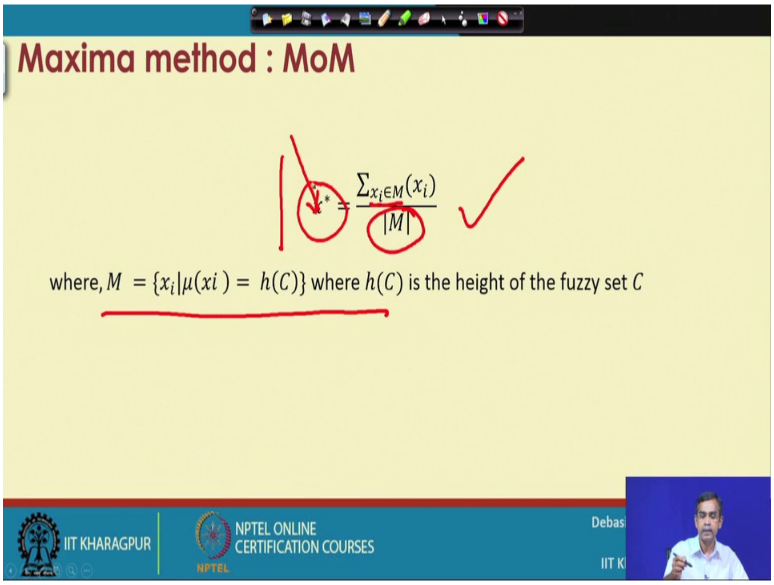
(Refer Slide Time: 05:27)



So, for example, again the same graph if you see. So, here the these are the different elements which has the highest value and the largest elements this one. So, these become the crisp value for this fuzzy set.

Now, one thing we can note that different method if we follow, so far we have discussed about the height method. Height method and first of maxima method is basically same if it is a unique anyway. So, all the methods if we follow they give the different result for the same input in fact. So, this means that result can vary from one approach to another, but all results are acceptable.

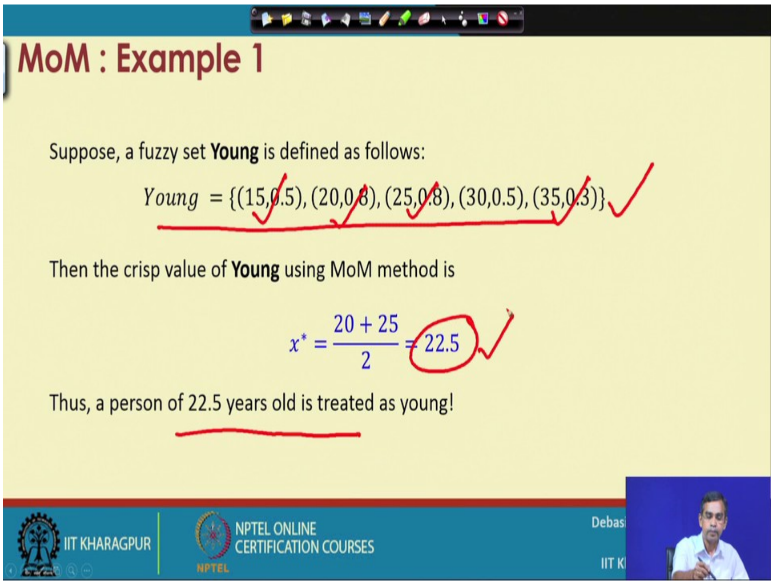
(Refer Slide Time: 06:12)



Now so this is the another method is called the mean of maxima method. So, in this case if we have more than one element having the highest value of their maxima then which will be the as crisp value. So, it basically takes the average of all the values that is there for which the height is more than 1. So, as an example this is basically expressed in this form where M; M is the set of all elements which has the membership value its same as the height of the fuzzy set. And then for all element x_i that is in M, we have to take the summation.

So, it basically this way we can take the mean of the maximum. Here M is basically size of the set n which has the membership value same as the height of the fuzzy set. So, this is a simple formula although it can be displaying the graph, but in the numerically if we have a fuzzy set and then we can use this expression to calculate the crisp value where $x^{¿}$ is the crisp value for the fuzzy set.

(Refer Slide Time: 07:33)

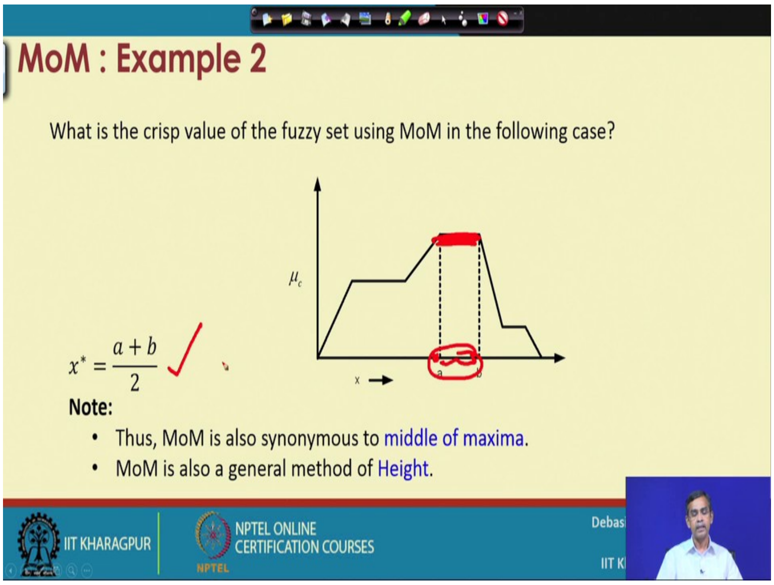


I would like to give some examples so far the min or maxima is concern one example.

Let us consider this is the fuzzy set, and if we follow the first of maxima method then this is the crisp value. If we follow the last of maxima method then this is the crisp value. And if we follow the mean of maxima method, then we see in these sets these are the two element which have the highest heights or have the membership values same as the height. So, taking the average of the two values and we can use we can get it that this is the crisp value for this fuzzy set. So, a fuzzy set is like this then the crisp value for this fuzzy set can be obtain as 22.5. For example, if this fuzzy set denotes the young as a fuzzy set then the person of the year 22.5 years old is treated as young.

So, for the crisp value is concerned, but according to the fuzzy 15 is also young 35 is also young 20 is also young 25, but according to the crisp the 22.5 is the young.

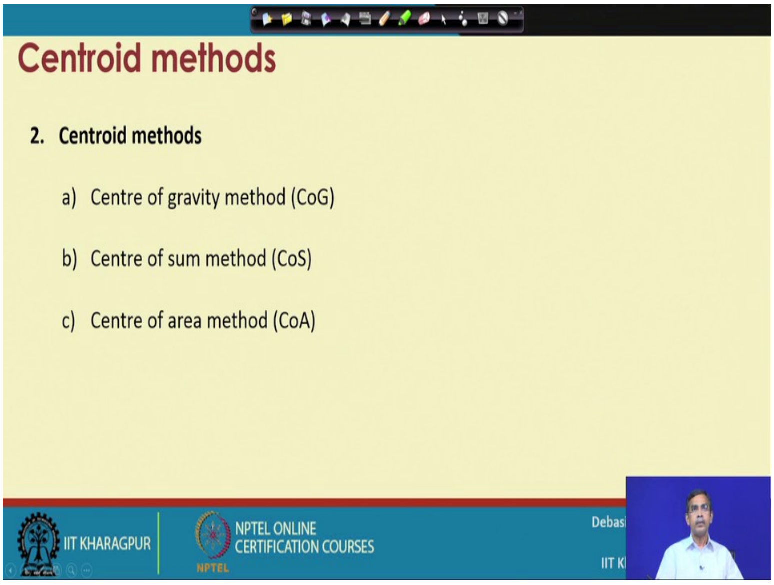
(Refer Slide Time: 08:59)



Now this is the another example, here if it is basically if you see the is a is a continuous sets right. So, these are the so many values which belongs to this are basically having the membership value same as the height. Then the average of this according to the mean of the method can be obtain as $\frac{a+b}{2}$. So, this is the a and b and this is basically average of all the values which belongs to in this range, then this is the crisp value. So, crisp value. So, crisp value in this case if we say that this is the a and this is b then crisp value.

That can be obtained as in the min of the method is this one. Now here actually this becomes a middle of maxima; because $\frac{a+b}{2}$ is basically middle of this one. So, sometimes min of max also it is called the middle of maxima method and in fact, this is the one method it is a generalise a generalised method, far the maxima method is concerned.

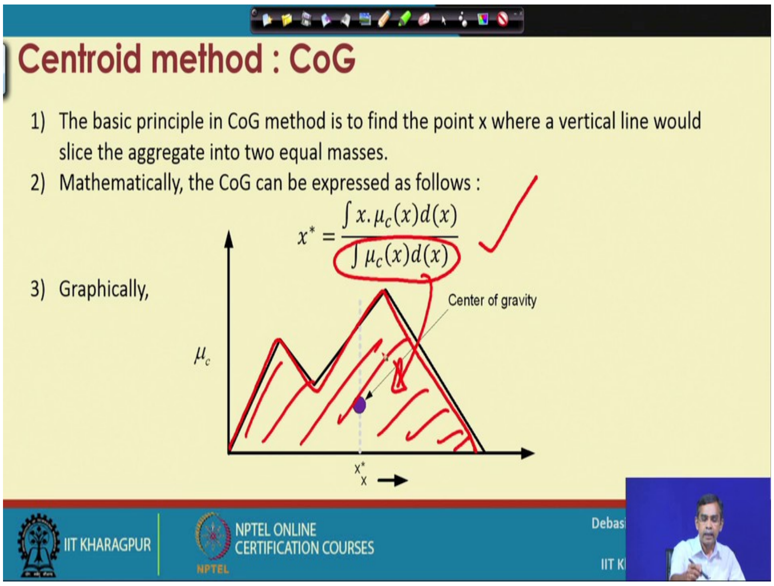
(Refer Slide Time: 10:06)



Now, let us discuss about the another approach it is called the centroid methods. Belong to this category there are methods centre of gravity method, centre of sum methods and centre of area methods, we will discuss each method one by one. This method; however, compared to maxima method is computationally much expensive; however, it gives more result better result than the maxima method.

Now, first we will discuss about centre of gravity method.

(Refer Slide Time: 10:40)



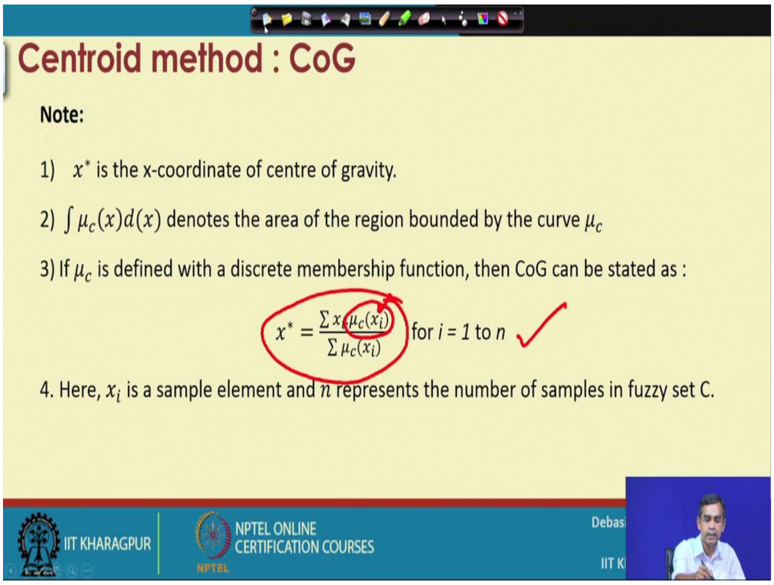
This method is popularly called CoG method. It is a short form of the centre of gravity method. Now we know exactly for a any for any geometrical object what exactly a centre of gravity means, it is basically similar to the centre of mass calculation of an object a geometrical object whether it is a 2-dimensional object or three dimensional object whatever it is there. So, it is basically centre is a centre where a vertical line can segregate the things into a two equal size of masses. So, that is a general concept of centre of gravity; that means, the entire mass will process through this point sort of thing.

Now so far the geometrical object is concern, the same thing looks like this. So, if it is an object then centre of centre is basically this one. So, this is basically centre of gravity sort of thing. Now the same concept it is applicable here, now in this graph this is a graph of a fuzzy set and this graph is look like this and the centre of mass suppose it lies here. Now if we draw a vertical line from this here on the x axis, which cut at this point this point is basically called the centre of mass point. Now having this centre of mass point, this basically the crisp value for this fuzzy set. Now so far the computation is concern how this can be calculated.

So, there is an expression for calculation of doing this thing assuming that. This x varies over a continuous range of values, and the graph of the membership values for this fuzzy set is like this, then the method by which the CoG of this set or a graphical thing can be calculated within this formula. So, it is basically $\frac{\int x.\mu_c(x)d(x)}{\int \mu_c(x)d(x)}$. You can note that these basically represents area of the curve under the curve μ_c. So, if this is a μ_c. So, these basically area of these portion. So, it basically represents this one and the numerate this portion is basically is a weighted. So, x into a particular area of this portion or x into corresponding $\mu_c(x)d(x)$, and then next x into another $\mu_c(x)d(x)$ and this one. So, if you take the instigation of all over the things it will gave the coverage of all the entire portion.

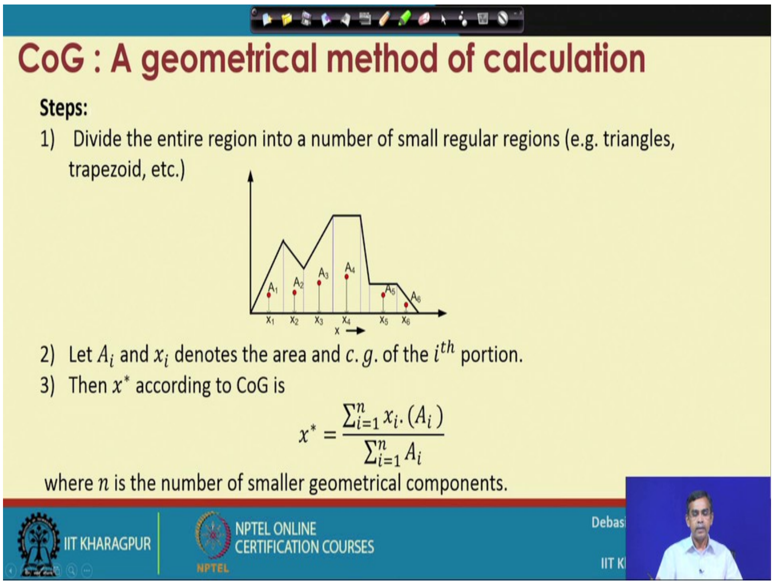
So, this is the formula that will be used to calculate the centre of gravity and the same formula can be also extended in case of discrete value also.

(Refer Slide Time: 13:35)



So, in case of discrete value the formula is like this. So, instead of integration we can use a summation formula if we know the value of $\mu_c(x_i)$ for different x_i mathematically, then mathematically this can be calculated easily. So, this is applicable for if the fuzzy set has the discrete set of elements and the previous example that we have discussed if it has the value for the continuous elements.

(Refer Slide Time: 14:11)



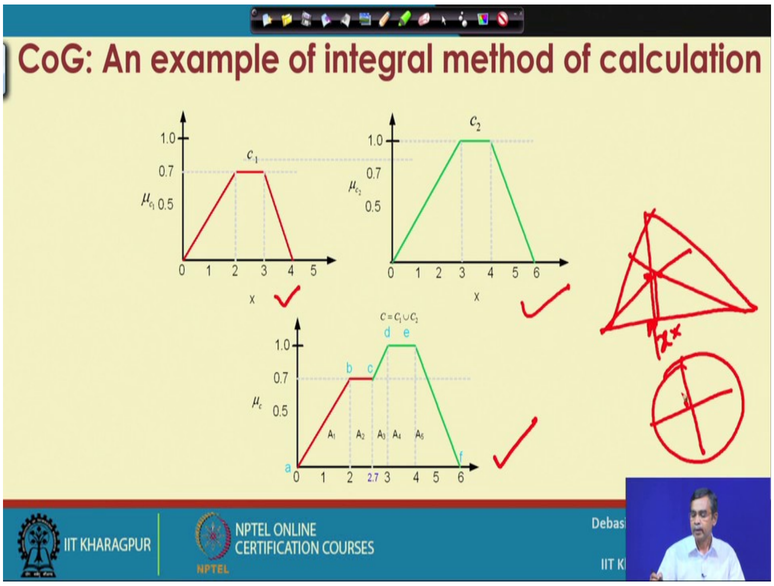
Now, this is one example how the CoG method can be applied to calculate manually, and I will give another example how the same method can be calculated numerically.

Now suppose, this is the one fuzzy set that is shown here and. So, the in this fuzzy set, we can do one thing we can have some segmented area, so that the area of each segments can be easily calculated. Now for example, if the this is the entire we can find one area of this one and then next area this one and then next area this one and so on so on. So, different portion of the area if we can indentify manually, then for each area applying a same method CoG formula, we can calculate its centre of gravity and then taking the sum of each then we can obtain this one. So, alternatively the method also can be like this one if we say this is the area of A_i -th segments.

And x_i is its centre of gravity, then taking the sum for all the centre gravity and their corresponding area and dividing by the total area of the curve, the we will get the crisp value for this fuzzy set. So, this is the one geometrical method by which the centre of CoG can be calculated and hence the crisp value of this can be calculated. So, this is a graphical method. Now for each graphical segment, this is just simply using are area of triangle we can calculate, and this is suppose using area of a trapezium we can calculate. So, calculation will not be tedious only the thing is a number of more calculation is involved, because there are n number segments then we have to calculate n areas and then there is a product and then average divided by the total area like.

Now, I can give an example of the same method but say is in numerically.

(Refer Slide Time: 16:23)



Now let us see these graph again here. So, suppose this is the one fuzzy set, this is another fuzzy set and this is another output of the fuzzy set. So, output of the fuzzy set is basically taking the union of the two and. So, if we take the union of the two and plot on the same graph it will give this one. Now so for the geometrical method is concern, again we can do the segmentation. So, it has A_1 , A_2 , A_3 , A_4 and A_5 . So, the five segments and for each segments, we can calculate how the area of the each area of each segment can be calculated we have an idea and also the centre of gravity.

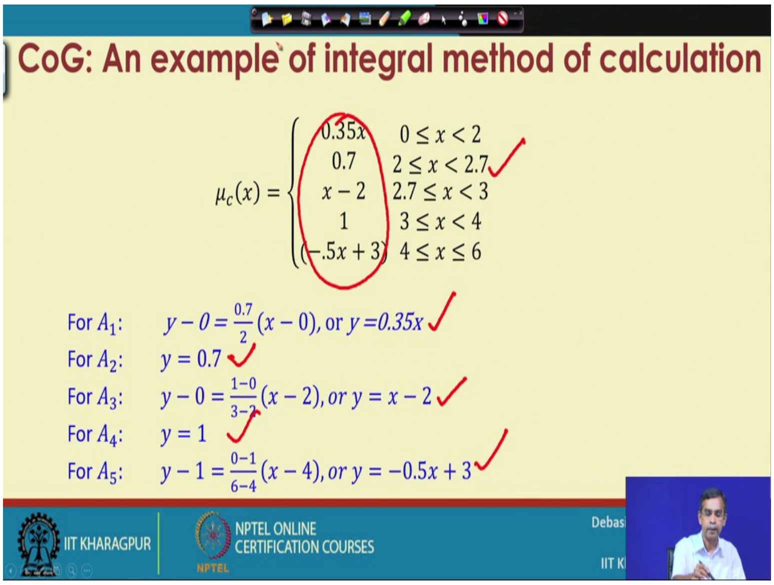
One thing if this is the one triangle I just forgot to mention it, then how you can calculate the point here which is the centre of gravity and therefore, we take the this one as the what is call the centre of gravity point say $x^{¿}$ like. So, idea is that if we take some geometrical method of this one and then this one who is basically point we intersect this one. So, it basically the centre of gravity for the circle it is very simple. So, it will. So, this one. So, this is the centre of gravity and like this one.

So, there is a geometrical formula by which the centre of gravity can be calculated area can be calculated accordingly. Now let us consider the same, but with the help of some numerical calculation. Now so we can calculate the area if we know what is the equation of this curve.

Because it is basically $\mu(x)d(x)$ formula if the equation is defined by this one now this is suppose a straight line. So, the equation of this line can be easily obtain if we now the slope of this line. And similarly this is the area, area can be obtained if we know this one and this is the line. So, this is the straight line into the $d(x)$ this one. So, eventually the idea is that if we know the different portion and their corresponding equation mathematically, then taking the simple numerical method of equation integration we can find the area of each pieces each piece and then the area of the entire curve and then the CoG can be can be roughly can be taken as the middle of this point then.

So, it is a basically little bit not so, much accurate whether inaccurate calculation, but this can be useful for it. Now let see detailed example about that how taking this into this using this information how we can calculate numerically.

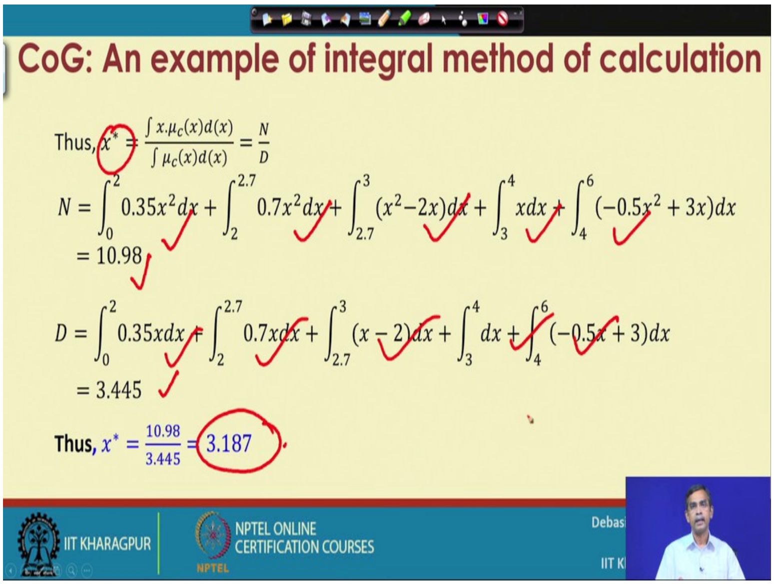
(Refer Slide Time: 19:11)



So, here is the idea the $\mu_c(x)$ that is the membership function of the output function can be expressed using this expression. This can be obtain readily for A_1 for example, as it is passes through (0, 0) and having the slope this one. So, we can this is the slope of the line and it passes through (0, 0) the first line rather. So, again we can see it. So, this is the area of the curve A_1.

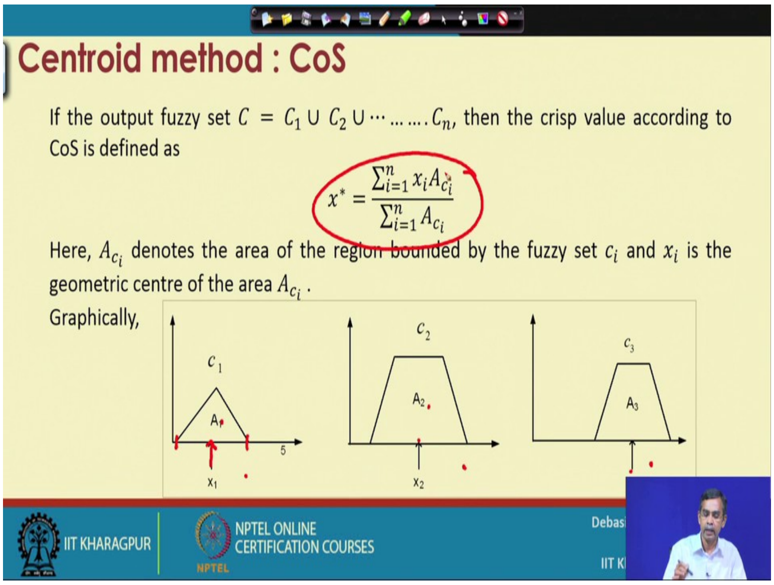
Similarly, area of the curve A_2 is like this which has the equation this one and area of the curve A_3, A_4 and A_5 can be calculated having these are the equations of the membership function it is there. So, this is basically can be. So, this this way area of the each parts can be calculated using some numerical form.

(Refer Slide Time: 20:09)



Now having this is the area then we will be able to calculate the CoG value for this fuzzy set, and this is the formula that we have already discussed. Now you can have shown here separately now the numerical component and then denominator component can be calculated. So, for the numerator component it as the different parts, for each piece actually. So, this is for the A_1 this is for the A_2 this is for the A_3 this is for the A_4 and this is for A_5 . So, the numerical result that can be obtained using this integration method is this one. Likewise, for the denominator for the five parts, which is shown here the value can be obtain this one, and therefore, CoG $x^{¿}$ can be calculated as this one this one. So, this means that the output fuzzy set for the output fuzzy set it has the corresponding crisp value according CoG method is like this.

(Refer Slide Time: 21:16)



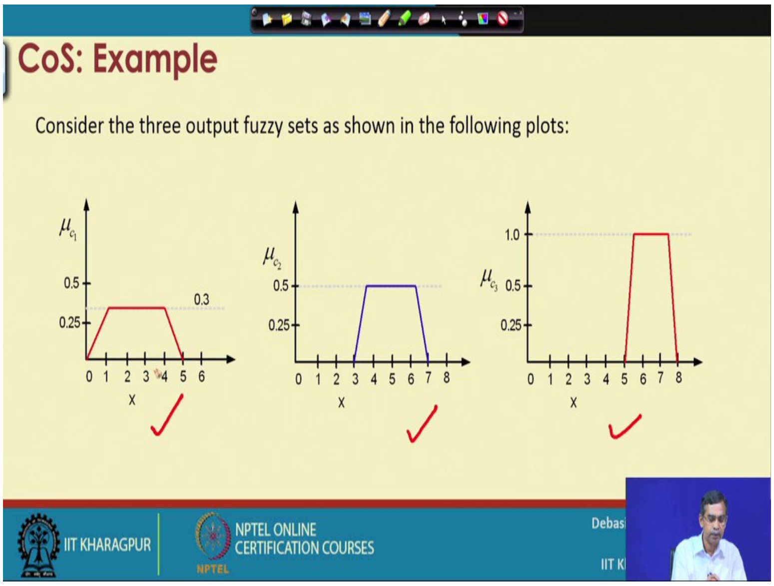
Now. So, this is an example how using an integral method, the CoG can be calculated. Now I want to give another example for the another method which belongs to the centred method it is call the centre of sum method. |It is relatively computationally very easy compared to the previous method CoG.

Now this method can be better explained if we consider let C the output fuzzy set obtained from the n number of fuzzy tests $C_1, C_2, \ldots, C_n$ etcetera. Then according to this method according to method the fuzzy crisp value the crisp value for the fuzzy set C can be obtain using this formula, where x_i is basically the middle value of the fuzzy set and A_{c_i} is basically the area of the fuzzy sets c_i and this is basically the sum of all areas. Now as an example suppose this is the c_1 and this is c_2 and this is c_3 and C is the output fuzzy sets. So, in CoG method we have plotted the three graphs together, but here we do not have to do these things rather we can take individually one by one.

So, the first one we can calculate the area A_1 it can be calculated either within geometrical method or within some numerical method, and x_1 is basically middle of the two; that means, this is the middle. So, x_1 into A_1 and for this x_2 into A_2 and x_3 into A_3 is the numerator component and $A_1 + A_2 + A_3$ is the denominator component then the co method CoS method will give you the crisp value for this fuzzy set. So, this is similar that of the CoG method, but in case of CoG method we have to plot all the graphs together and then taking the resultant graph and then for the resultant graph we have to calculate CoG, but here we do not have to do.

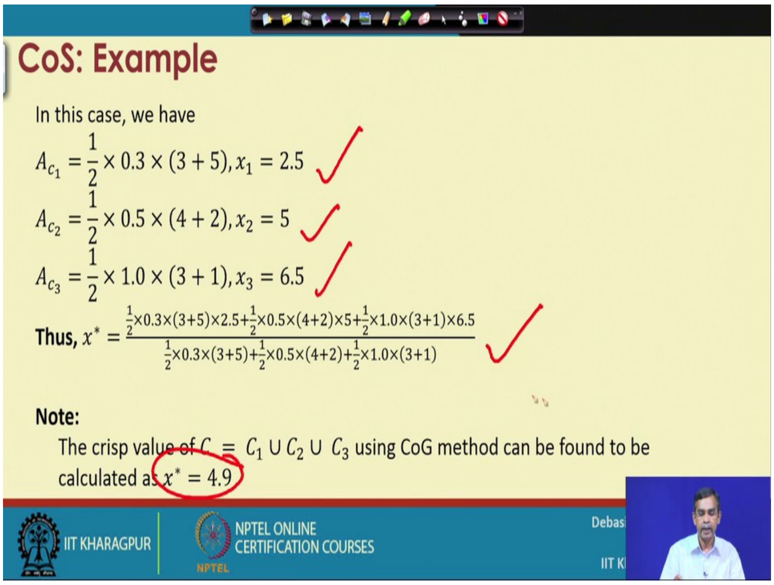
We have to take on an individual output and then take the summation of all those things and then average, and then the result can be obtained. So, result definitely will be different than the CoG method if course, but it is the computationally less expensive.

(Refer Slide Time: 23:34)



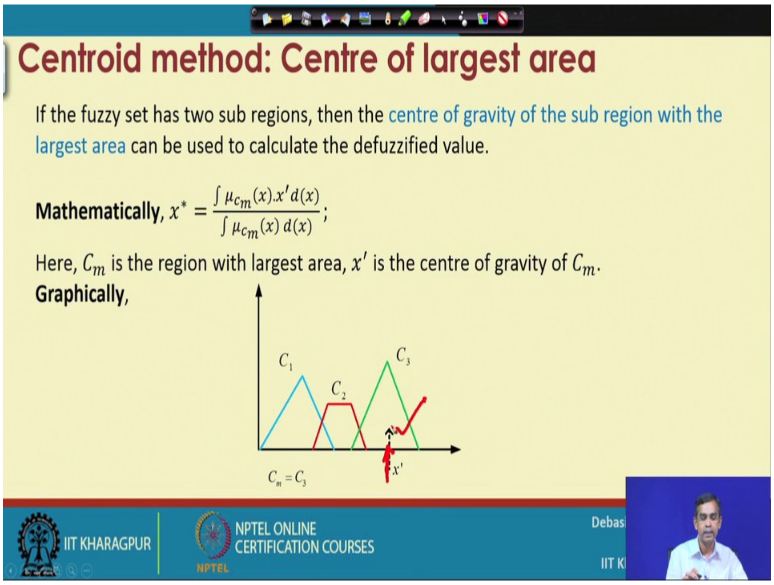
Now, so this is the ca method and this is an example we can again exercise. So, this is the one curve c_1 this is the c_2 and c_3 . So, for this c_1 we can easily calculate the area of this one and this one let see, what is the area of the 3 components here and we can calculate.

(Refer Slide Time: 24:01)



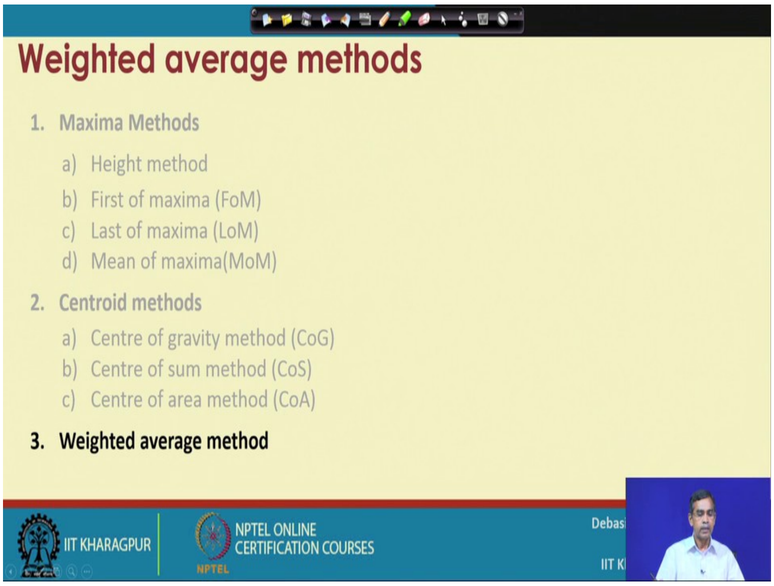
So, we can calculate here for example, for the first fuzzy set c_1 this is the area second and second and using the formula we can obtain the crisp value for this, and this is the result this one. Now again if we apply if we apply the CoG method to the same graph; obviously, the result will be different it can be observe that result that the CoS method will give little bit higher values than the CoG method, because it will take area for the two curves more than twice whereas, the same area will taken only into one in case of CoG method. So, these are the CoS method and there is another is the simplest method it is called the centre of largest area.

(Refer Slide Time: 24:45)



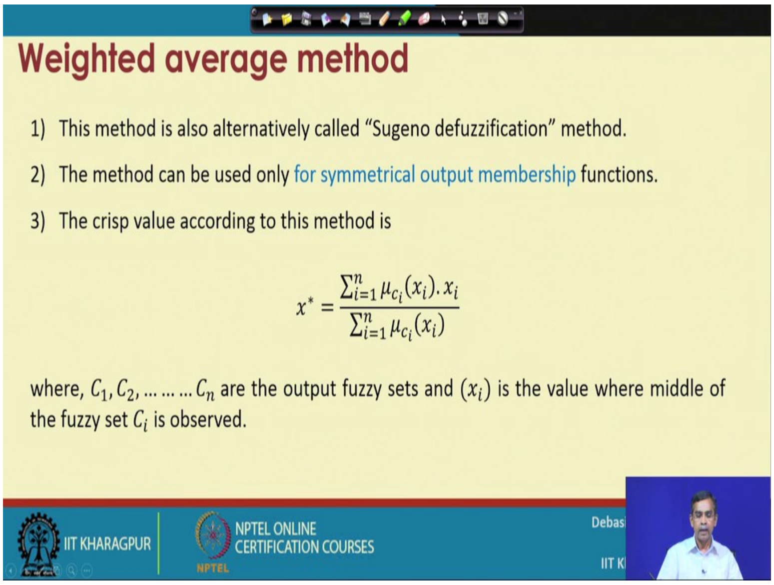
It is just one another what is called the simplification of the centre of sum method rather it will rather consider only the one fuzzy sets which having a largest area. So, if this is the fuzzy set having a largest area then it will take only that fuzzy sets and then it has the area of this one and this one divided by x. So, it basically gives this is the crisp value of this fuzzy set. So, this method is very simplified form of the previous method hardly it is used, but the mostly used method is CoG and then the co then the CoS method is preferable yeah.

(Refer Slide Time: 25:29)



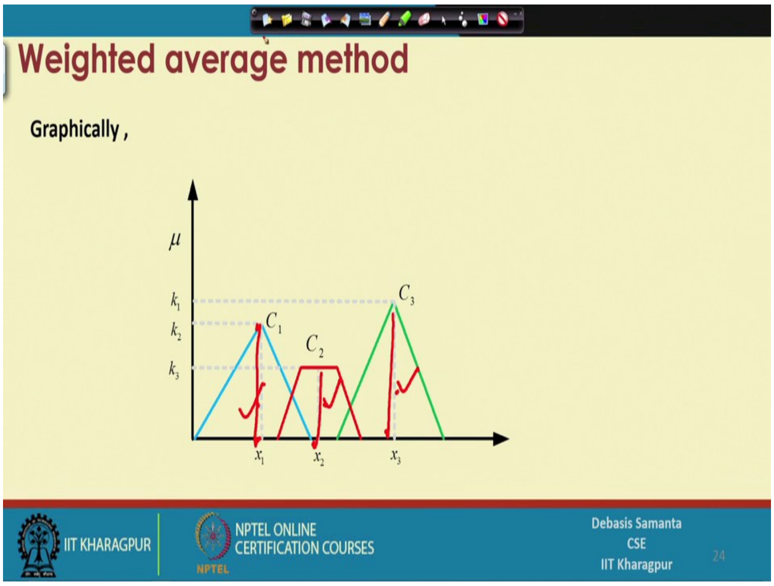
So, this the different method that we have discussed about and then weighted average method I will just discuss quickly.

(Refer Slide Time: 25:47)



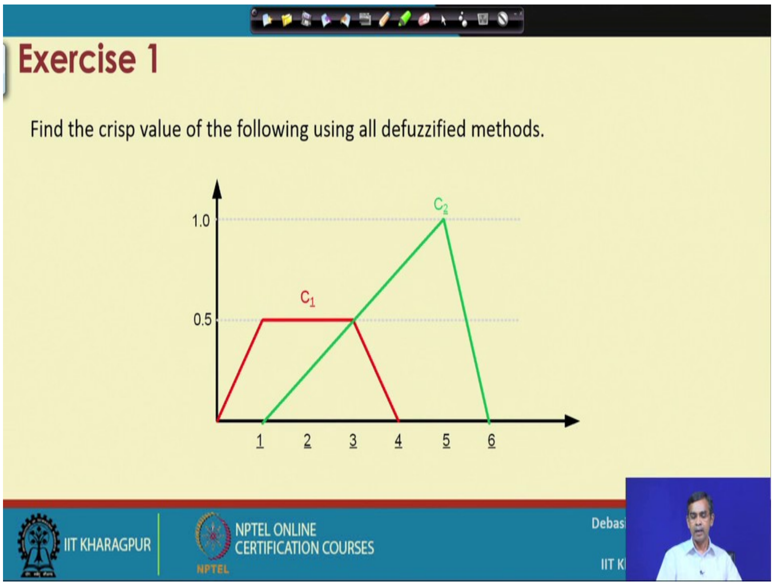
So, it is very simple to know and. So, the weighted average method it is similar to that one. So, similar to the this one this method also called popularly called Sugeno defuzzification method and; however, this method is a simplification of the previous centroid method, but it is it is only applicable for the symmetrical output membership. That means, if a fuzzy set has the symmetric in shape then only we can apply this method symmetric means so, these are the symmetric method.

(Refer Slide Time: 26:18)



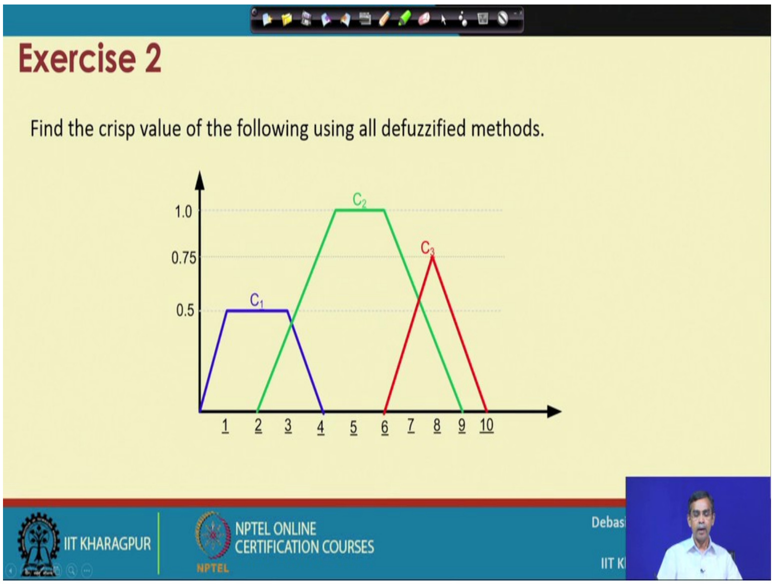
And then we can have it this is a symmetric, this is all the curve this is also symmetric this is also symmetric and this is also symmetric. Then for this symmetric method we can have the middle value of this x_1 and this is x_2 and this is x_3 they taken individually and then area of this one area of this, one area of this one divided by this one it is there. So, it is basically same as the CoS method or CoG method we can say in some extent, but it is applicable only for CoG if it is for if we can use it for the symmetrical fuzzy set then it gives a better result that is why if we know that the fuzzy sets are symmetric, then without any second thought we can use for this method and then we can get it. Now in the last few slides I have plan few examples so that you can understand it.

(Refer Slide Time: 27:11)



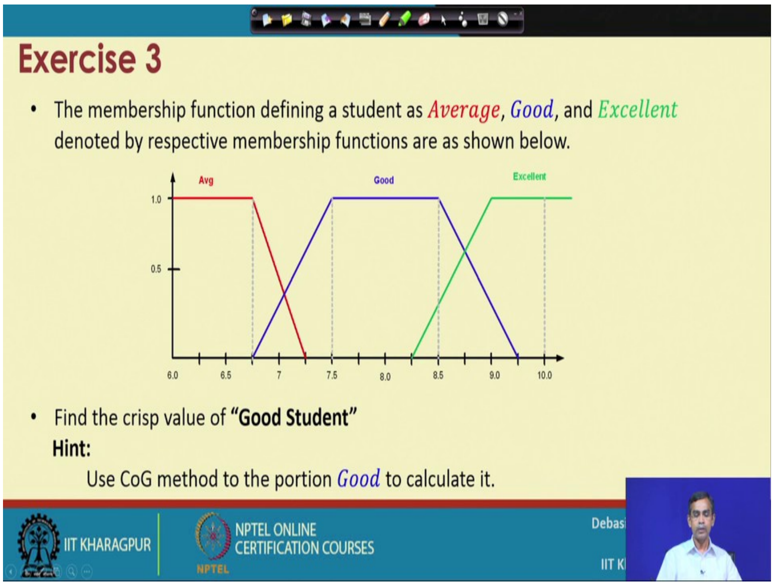
For example, this is the one output having the two fuzzy sets c_1 and c_2, we can easily calculate the either using maxima method or CoG method or CoS method. So, you should try using the different method, how the crisp value can be obtain and you can compare the results easily.

(Refer Slide Time: 27:38)



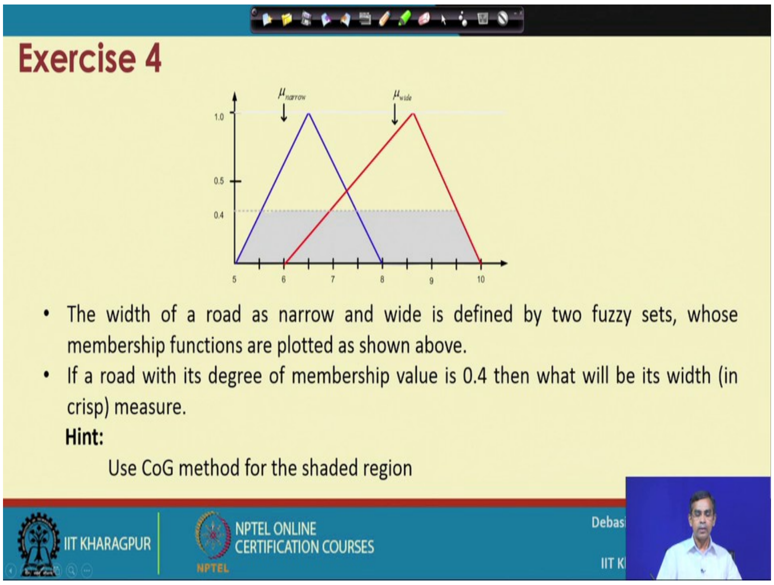
And this is the another example. So, we can find defuzzified method following either maxima method or CoG method or CoS method and then weighted method.

(Refer Slide Time: 27:56)



Now if I ask you how the crisp value for the good student whether this is a fuzzy set for the three set is given, but as the good is the only our objective. So, we can limit our fuzzification to this portion only and then we can calculate. Again the same method either centroid method or maxima method or weighted method can be applied to calculate it easily then you can understand what is the crisp value corresponding to this fuzzy set.

(Refer Slide Time: 28:29)

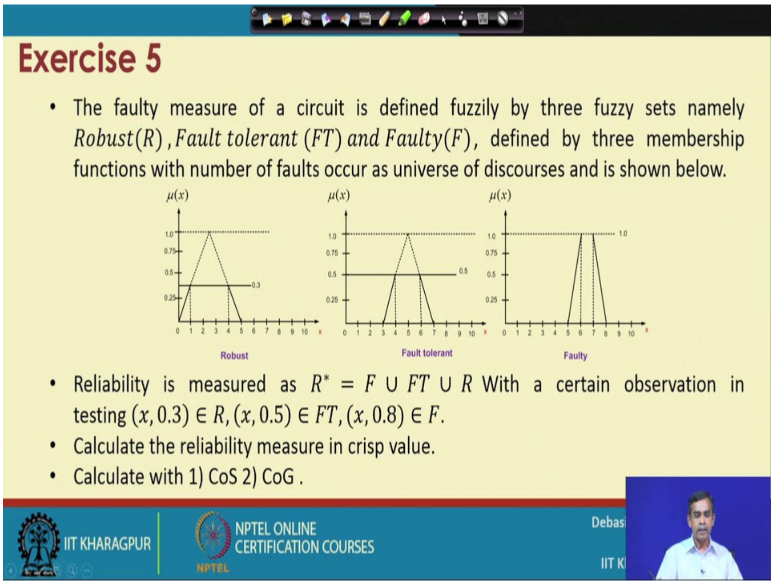


Now, this is an another example. So, here the two fuzzy sets namely the narrow of a road and wideness of a road is given. So, the fuzzy sets are described for the narrow this one and then wide fuzzy set is this one now here. So, suppose here actually the width of a fuzzy set having different what is called the degree is known to you, then we have to calculate what is the road if its degree of membership value is 0.4.

So, we can the simplest is that the two graph can be plotted on the same plots and then taking the common area corresponding the qualified value of the membership value. For example, in this case the qualified value for the membership is 0.4. So, if we take this curve 0.4 and then this is a common area and so, we have to take the fuzzified value of this is the fuzzified value of this result. Now we can take the crisp value taking again CoG method or CoS method or maxima method or some weighted method and then we can calculate the crisp value. Crisp value basically if the road is narrow and wide is defined by some fuzzy sets then for a particular road having some width.

And its degree of membership 0.4 then the crisp value can be obtained. So, the area and then corresponding the crisp value gives that if the road is having what is call the width say some value and its degree of membership 0.4, then this basically gives you the crisp value.

(Refer Slide Time: 30:28)



Now there is another example that is more interesting to note; say suppose here the faulty measure of a circuit that is that can be defined fuzzically fuzzily by three different fuzzy sets. There are three fuzzy sets namely robust, fault tolerant and faulty. So, these are the three fuzzy sets and corresponding the membership value is defined for this robust and fault tolerant and then faulty.

Now, suppose reliability is measured this is basically reliability whether it is a faulty or fault tolerant or a robust. So, it basically reliability of a system is measured by this formula. Now if we define all these are the fuzzily then their resultant value is also can be obtain fuzzily. So, union of the three fuzzy sets can give you the reliability of the fuzzy sets.

Now the same thing can be plotted on the same graph and then we can have the reliability measure. As an particular instance say suppose one circuit is tested with some x value and then degree of membership is 0.3 that is basically belongs to the robust and then x is the number of circuit fault that is obtained with degree of membership 0.5 and this is the belongs to with the fault tolerant, and there is a x is the number of test performed with degree of membership which basically gives the faultiness.

So, we can obtain its crisp value; that means, crisp for the reliability if we take this is the output for the first component; that means, robust and this is for the second fault tolerant and this is the area covered by $\mu = 8.0$. So, it is 8.0. So, this is the area and then we can take either CoS method or CoG method and then we will be able to calculate the crisp value and that basically the crisp value for the reliability of the circuit. So, these are the few example that we have discussed and so this way the defuzzification method for the different according to the different techniques can be obtained. Now in the next lecture we will apply this defuzzification technique in more general sense whenever we will discuss about fuzzy system design. So, our next topics will be how we can design fuzzy sets using the different concepts that we have learn so far.

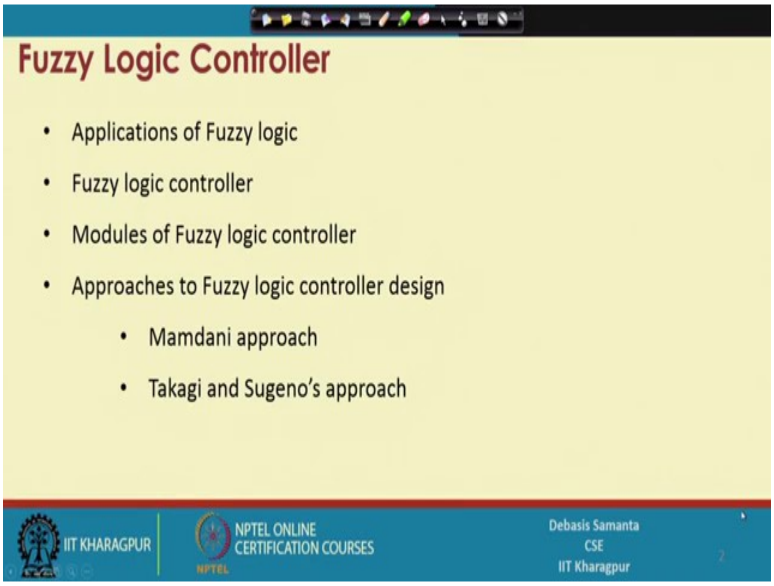
Thank you we will meet again in the next lecture

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 11
Fuzzy logic controller

So, we have discussed about how the different operations related to fuzzy elements can be carried out. Now, we are in a position to discuss about designing a complete fuzzy system. Now, the fuzzy system that we are going to discuss is very popular in fuzzy application, fuzzy world and this is called the Fuzzy logic controller.

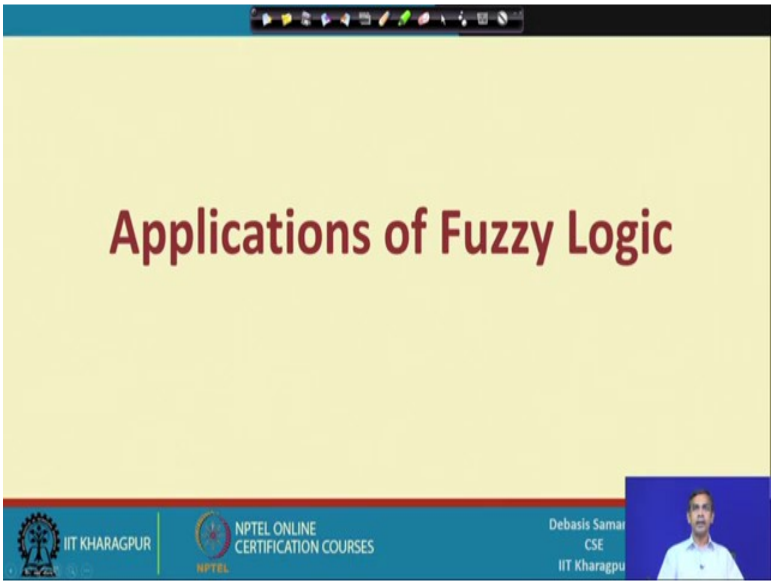
(Refer Slide Time: 00:45)



So, actually we are going to discuss about how a fuzzy logic controller can be can be designed and so far the fuzzy logic controller design is concerned they are broadly two approaches are known the first approach is called Mamdani approach and the second approach is called Takagi Sugeno approach.

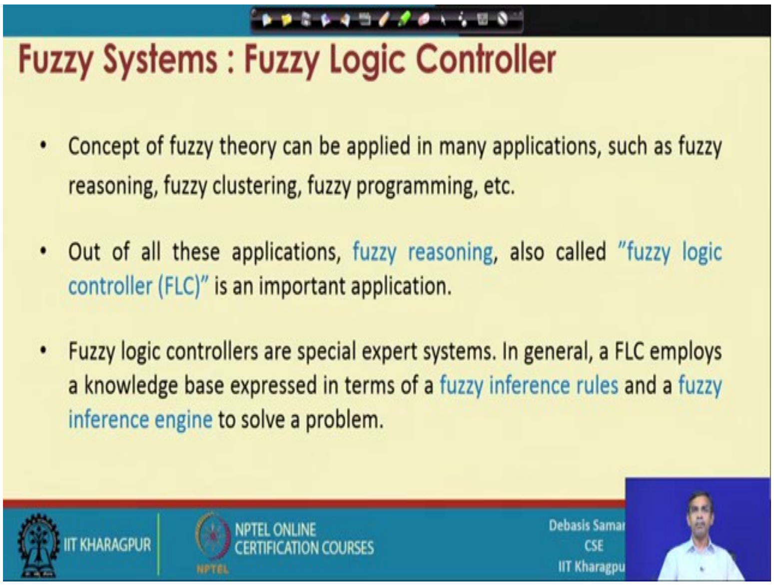
So, we will discuss about the two approaches in the first we will learn about Mamdani approach and then in the next lecture we will discuss about Takagi Sugeno approach.

(Refer Slide Time: 01:18)



Now, first we have to understand about that what are the different applications of the fuzzy logic. So, there are many applications of the fuzzy logic.

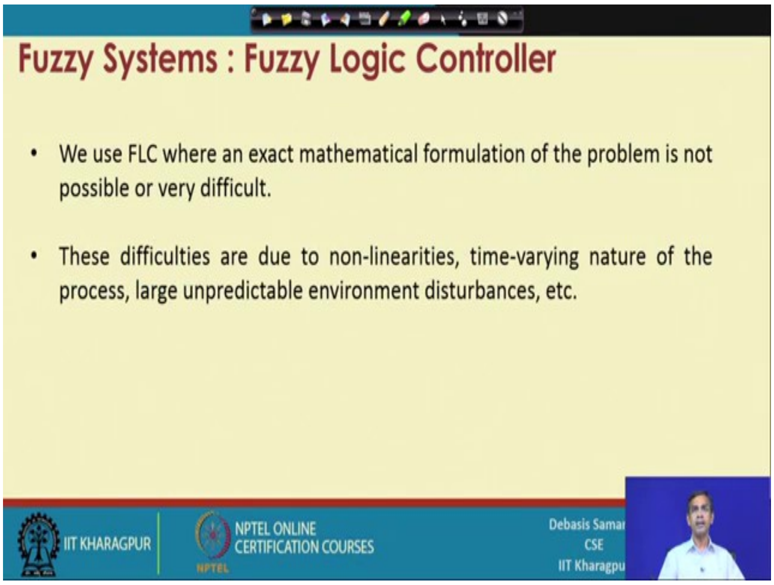
(Refer Slide Time: 01:24)



Few application that I have mentioned here one example is called a fuzzy reasoning, another is called a fuzzy clustering, fuzzy programming and so many. Now, out of these application, fuzzy reasoning this is also alternatively called as fuzzy logic controller is a widely used on application.

Now, fuzzy logic controller are the type of expert system is a special expert system we can say in general it employs a knowledge base or we can say fuzzy rule base and this fuzzy rule based is expressed in terms of a set of fuzzy inference rules and the fuzzy inference rule is used by one engine it is called the fuzzy inference engine to solve any problem. So, far the fuzzy logic controller is concerned or designing a fuzzy logic controller is concerned, the most important task that we have to carried out is that how a fuzzy rule based system can be developed and then how the fuzzy inference engine can be built on that fuzzy inference rule or fuzzy rule base. So, will discuss about these two things, first we will discuss about.

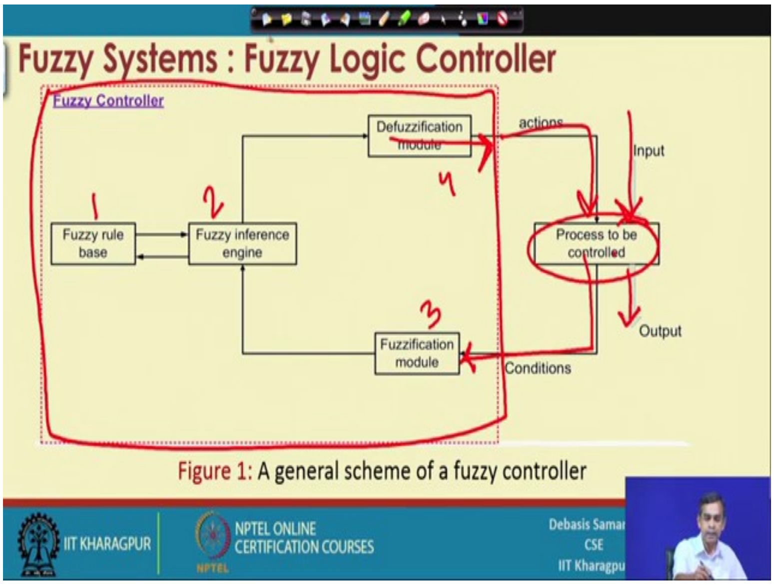
(Refer Slide Time: 02:34)



So, obviously, there are few problems where exact mathematical formulation of the problem is not available or is very difficult because of the many uncertainties are there. So, uncertainty may be due to non-linearities in the input; that means, the input does not vary with linear relation or it is a time varying or different time it varies the values or it has lot of noises due to the environmental disturbances. So, the value is really unpredictable at a time. So, if these are the situations having the input then we should follow the fuzzy system to develop and to solve this kind of problem.

So, fuzzy logic controller is one example here and we will see exactly how such a fuzzy logic controller can be developed.

(Refer Slide Time: 03:31)

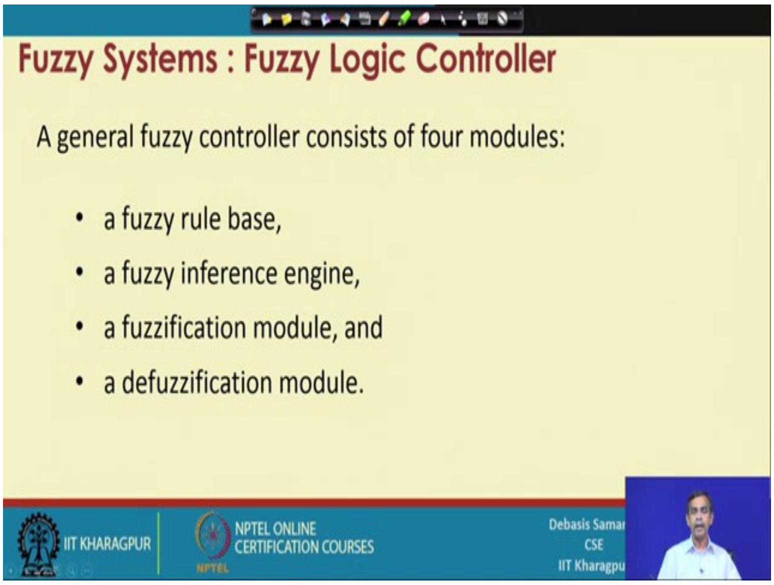


Anyway, for the fuzzy logic controller is concerned this diagram basically show the overview of a fuzzy controller system and if we see it carefully. So, the fuzzy controller this is the portion of the fuzzy controller right and this is basically an external interface with the outer world. So, these are the basically fuzzy world and this is basically the crips world we will come into this portion later on.

Now, in this fuzzy system you can see there are four basic tasks involved. So, the first is fuzzy rule base, second is fuzzy inference engine and third is fuzzification module and finally, fourth is defuzzification module. So, these are the tasks if we can plan it then a fuzzy system can be developed. Once, this fuzzy system is developed any input that is basically crisp input can be given to some controller. That means, the controller means it is basically the controller which will control with the eight of fuzzy controller actually and it will take any input and this input will go to the fuzzy system as a conditions. It will process it and then it will give an output which basically the crips output after defuzzification and this will be as an action. So, process will get output then this is the output that is has to be followed.

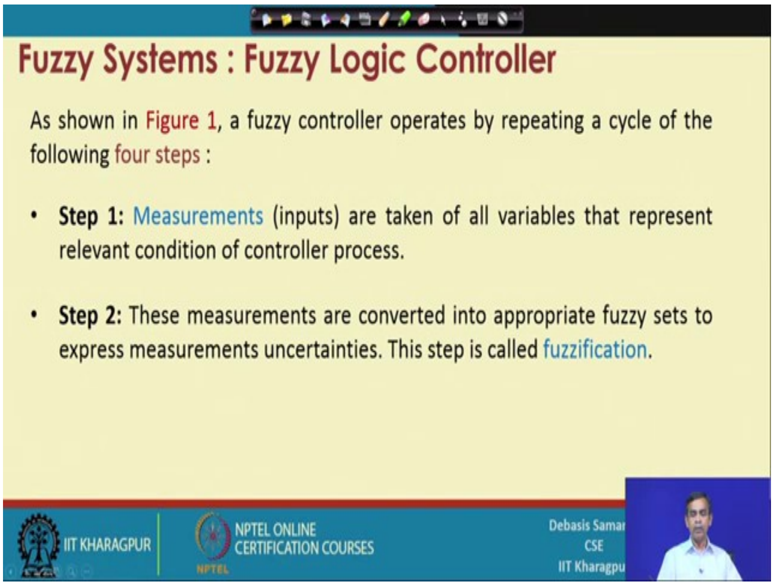
So, this is the basic idea about the fuzzy logic controller and we understood that there are mainly four different what is called the parts here the parts are fuzzy rule base and then fuzzy inference engine, the fabrication module and defuzzification module.

(Refer Slide Time: 05:27)



So, for the fuzzy logic controller or a fuzzy system reason is concerned we have to just design these four components then system will be built up.

(Refer Slide Time: 05:39)



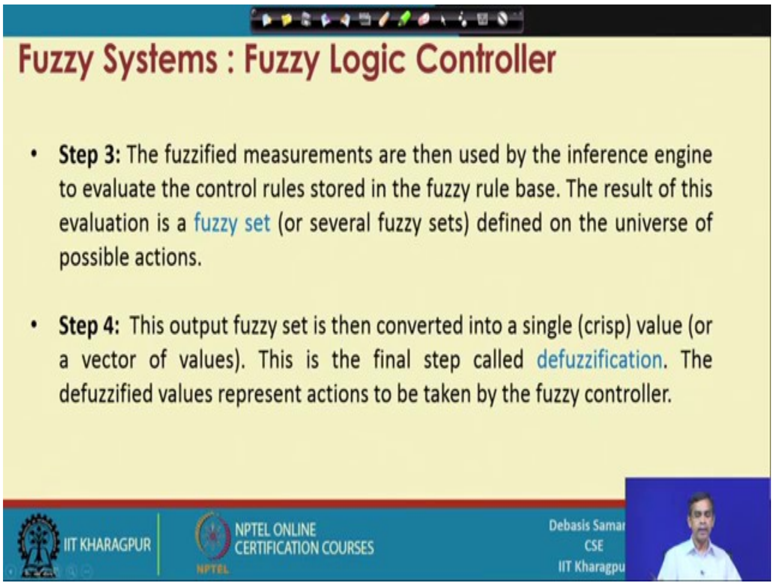
Now, I will just little bit a brief detail about the different processes involved. So, for the fuzzy logic controller is concerned, it is basically a cyclic process. It will take an input it will decide what exactly the control to be decided producing output take another input and output, is a cyclic method.

Now, we can consider on say AC, AC to be control if we change the temperature if we change the humidity these are the inputs suppose and then fuzzy logic knows exactly with the different temperature and different humidity what the output. So, far the motor rotation is come some it will produce an output we go to the motor rotator and then output will take this value and rotate accordingly. So, this way AC can be controlled air conditioned can be control like. So, this is an example.

Now, here the different steps that is involved there are mainly four steps. In case of the first steps, we have to take the input that is basically called a measurements it is basically for a system there may be one or more input. So, definitely we should consider all inputs taken together and that can be considered the condition to the controller process and then these measurement which is basically an input to the system is a crisp value. So, this needs to be fuzzified.

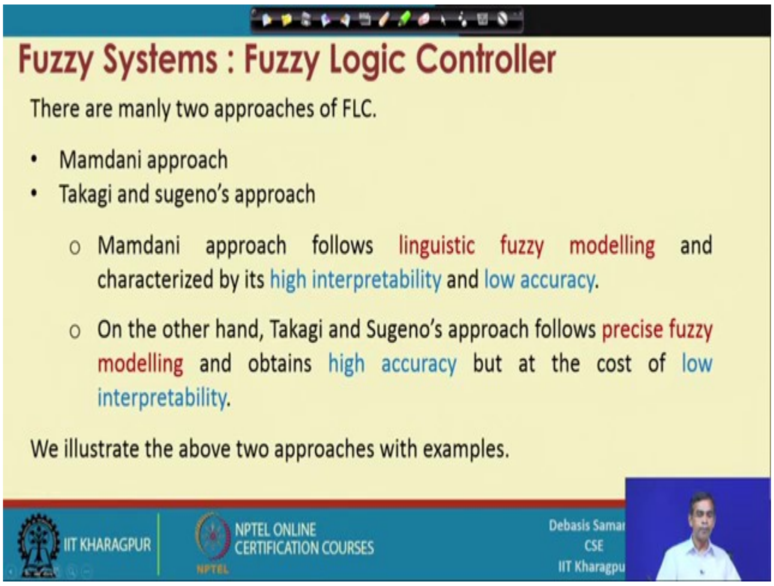
So, this second step is called the fuzzification and then one the major input is taken and it is fuzzified. The second, the third step is basically all these fuzzified inputs are to be used and then pump to the inference engine. The inference engine will basically evaluate what are the control rules to be followed that is basically they are in the fuzzy rule base.

(Refer Slide Time: 07:05)



The result of this evolution basically all provides a fuzzy set or maybe a several fuzzy sets and then output of fuzzy set or fuzzy sets can be considered and then taken as the output of the overall system. The output that we have obtained it is basically in terms of fuzzy set or fuzzy sets then we can convert the corresponding fuzzy set or fuzzy sets into a crisp value or set a vector of crisp values and this is called the defuzzification. So, these are the four steps that is involved in case of fuzzy logic controller.

(Refer Slide Time: 08:00)

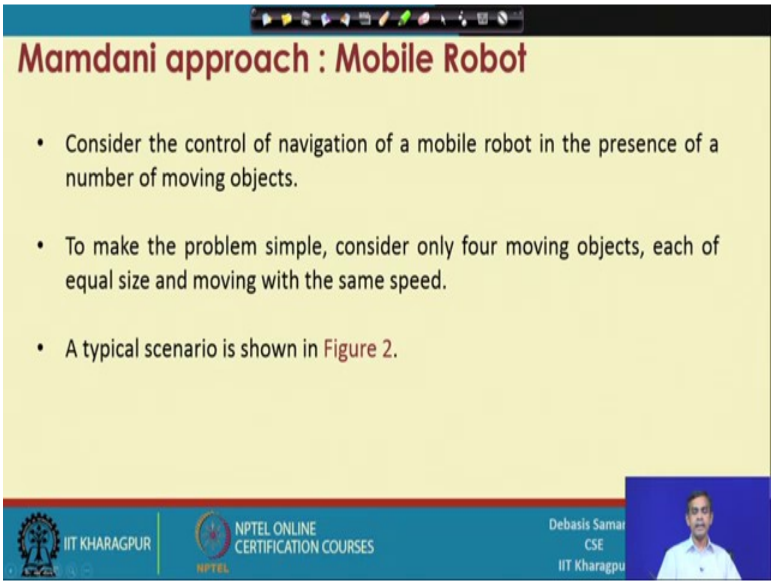


Now, let us come to the approaches there are two approaches I told you one is called a Mamdani approach and another is Takagi Sugeno's approach. Mamdani approach basically the simplest and more popular approach it is simplest because it is highly interpretable. That means, if we see the system we can or any lay man also can interpret. This is the concept of the system. However, it provides a little bit lesser accuracy compared to the Takagi Sugeno's approach.

And this model the Mamdani approach basic follow the linguistic fuzzy module. That means, all the fuzzy sets should be available to us some in terms of some linguistic states. On the other hand, the Takagi and Sugeno approach it follows precise fuzzy modelling it is more numerical than the linguistic fuzzy modelling that is used in Mamdani approach. In fact, compared to the Mamdani approach provides better result. However, it is low interpretable as it is mathematically little bit expressed. So interpretation is bit difficult for the general user.

Anyway, we will discuss first, will discuss first Mamdani approach and then the Takagi Sugeno approach with some case studies. So, that we can understand the concept it is there.

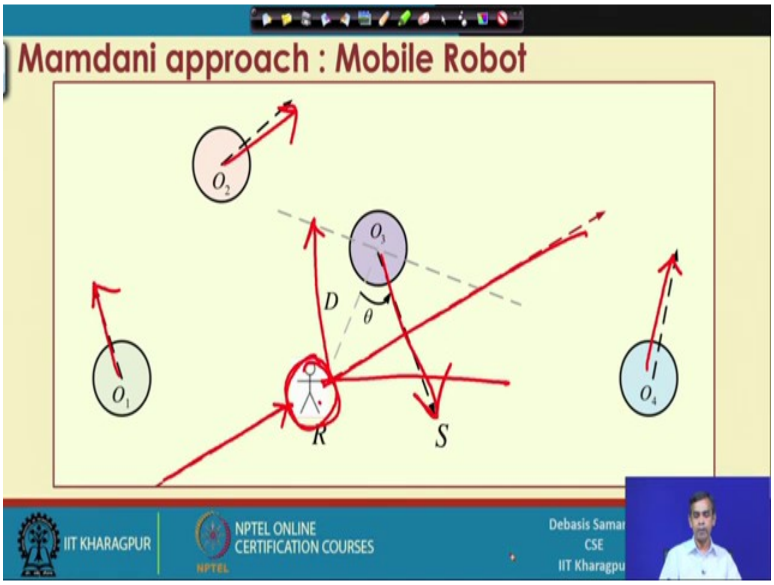
(Refer Slide Time: 09:23)



Now, we will discuss the first Mamdani approach and to discuss the Mamdani approach will consider an example it is basically movement of robot we can say it is the mobile robot. So, the problem here is a robot has to move in presence of several objects that is available there. So, robot has to move in such a way that it should avoid the collision and all the objects are not necessarily static objects they are also moving. So, it is a basically problem at any instant mathematically we do not know which path needs to be followed. So, path can be followed with lot of uncertainty or lot of variation in the non-linearity of inputs or time varying input and so this is a critical problem and this problem really very difficult to solve a simple programming approach.

So, we will see how such an application can be developed using some fuzzy theory, fuzzy logic. Now, a typical scenario of the mobile robot I want to give it fast and we have to discuss with certain assumption, assumption these that the robot has to move in presence of four moving objects and we also assume that each objects are of equal size and all objects those are there moving with the same speed. However, this is a simple assumption so that we can discuss it and then learn it, but these assumption not necessary to be followed in actual movement of the mobile robot consideration. So, anyway so that is an extension of this.

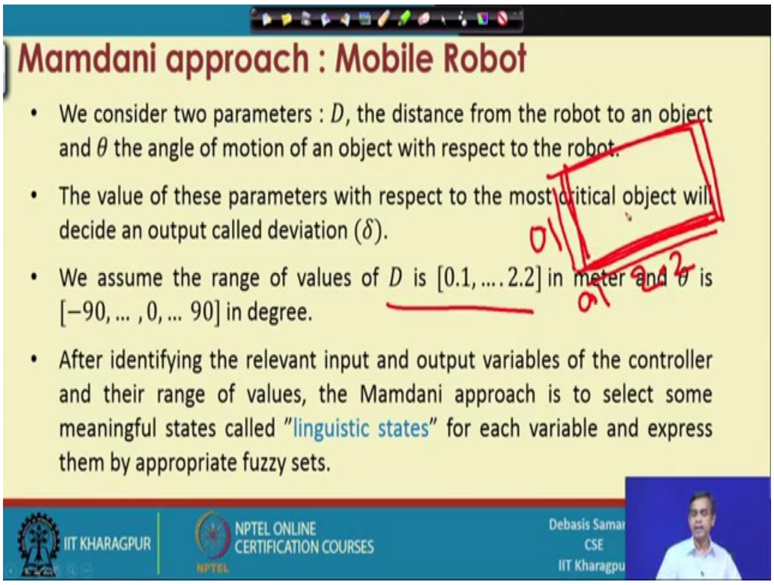
(Refer Slide Time: 11:10)



Now, this is the one simple display of the scenario of a particular instant and here assume that this is the robot has to move, and there are four different objects $O_{1,} O_{2,} O_3$ and O_4 in the vicinity of the robots at any moment robot is moving here in this direction and the different objects moving like O_1 is moving along this direction, O_2 is moving along this direction, O_3 is moving along this direction, O_4 is moving this term. Now, at any instant the robot has to take a decision if all these movements are there then which path he should follow, he should follow the same path or he will follow this path or in this path. So, it basically decide that as a next direction of the robot at any instant when he sees the different situation of the objects.

So, input to the robot can be obtained by some means about the different movement of the objects by some camera or whatever it is that is available this one and then it can calibrate and then different objects and his movement can be obtained. So, this is the case and then our task is basically to design a fuzzy controller for the robot, so that robot can use this fuzzy controller to take its movement direction at the presence of the different objects around it.

(Refer Slide Time: 12:38)

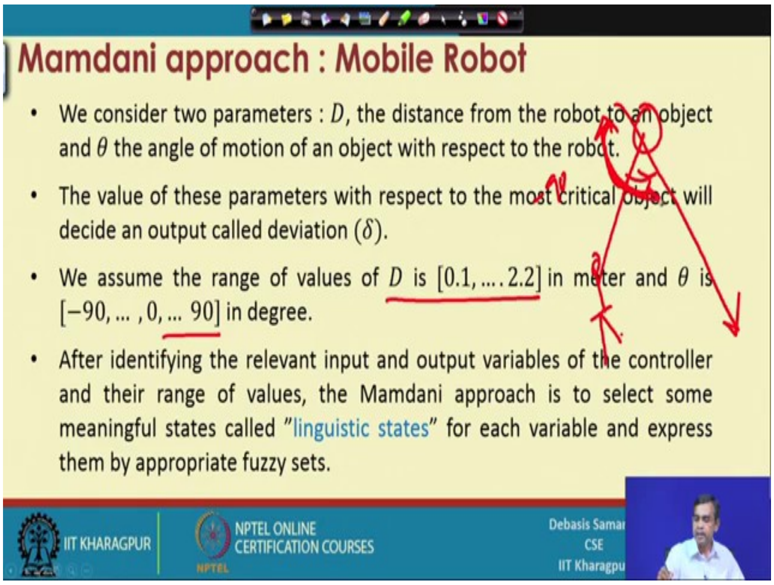


So, this is the application. So, this is the application. So, if we consider if we carefully observe this system we can consider that how the input to the system can be specified and then what is the output is there. Now, here as an input to the robot, we can consider two parameters, one is D the distance from the robot to an object and θ the angular motion of an object with respect to the robot. So, these are the two input that can be used to the fuzzy logic controller for the robot to decide it.

So, for the output is concerned we can decide one output it is basically called the deviation. That means, from the step from its own line how much deviation can be there. Now, here we will consider the first input D as this one here this means basically this basically signifies the total area of the movement of the robots. That means, it will start from the location 0.1 and 2.2 horizontal wise and vertical wise. So, it is like this. So, this is the 0.1 and this is at 2.2. So, this is the area along this one and this is the size of this one and this one. So, this is the total area. So, it is 0.1. So, 0.1 and this is 2.2. So, this is the area total by which the robot moment will be restricted.

So, considering this is the D. That means, range of values of the D that can be available like this and then the θ the rotation. That means, the angular direction of the different objects we consider in the range [-90 to 90].

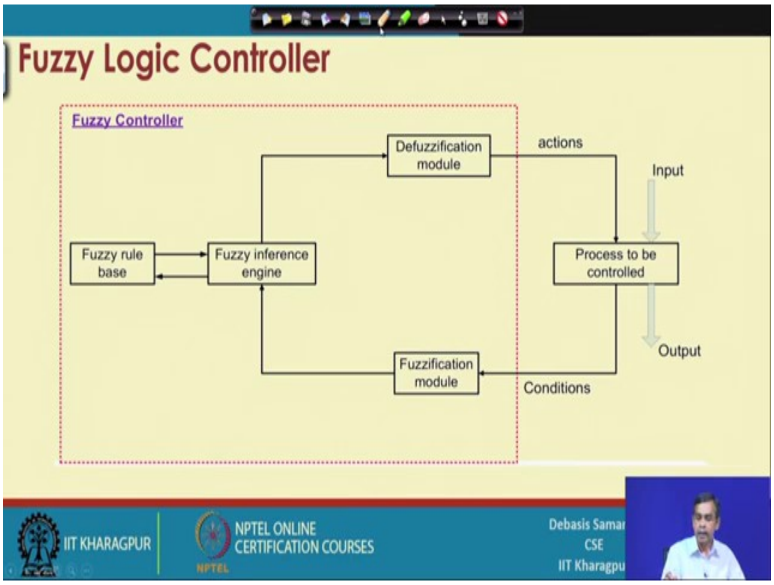
(Refer Slide Time: 14:21)



So, if this is the robot and then, with respect to this over the object that can be moved here right, if this is the movement and then according to this with the angle of this one, this can be if it is anti-clockwise then it is basically 0 to 90 degree and if it is clockwise then is -90 to 0. So, this is the range basically that the robot has to take a measure about an object with respect to this one. So, these are the two inputs and then output also similarly for this deviation will be -90 to 90. That means, toward left and toward right according to that.

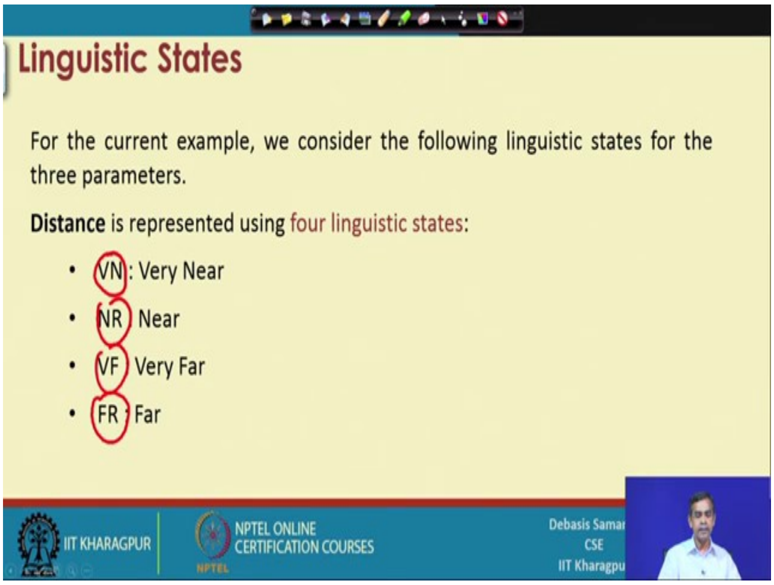
Now, the fuzzy sets that is required to describe this kind of behaviour and will express according Mamdani approach it basically decides the linguistic states for each input here D, θ and then output this one in terms of fuzzy sets. So, will discuss what are the different linguistic states for this particular example can be obtained. So, the linguistic state that can be obtained for this fuzzy set is given here.

(Refer Slide Time: 15:44)



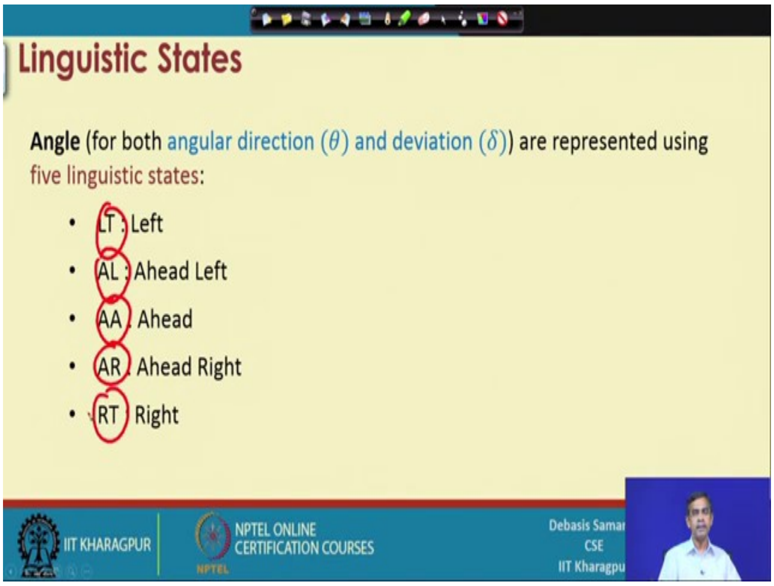
So, anyway we will discuss about first is the, these are the fuzzy controller system as we have already mentioned here. First, we have to decide about fuzzy rule base. Now, in order to decide the fuzzy rule base, we have to fix on the fuzzy linguistic states according to the Mamdani approach. So, first let us discuss about how the fuzzy rule base can be discussed and then we will discuss about fuzzy inference engine.

(Refer Slide Time: 16:08)



So, for the fuzzy rule base is concerned as I told you we have to first discuss the different input that we have to consider and then output how they can be expressed in terms of fuzzy sets. So, for this robot mobile robot, we consider for the distance D has the three the four different states; that means, linguistic states they are basically defining the distance as a fuzzy sets we discussed that four, one is very near denoted as VN, and the near denoted as NR, very far VF, FR far that mean distance can be fuzzily described as very near, near very far and far in terms of the four different fuzzy linguistics.

(Refer Slide Time: 17:00)

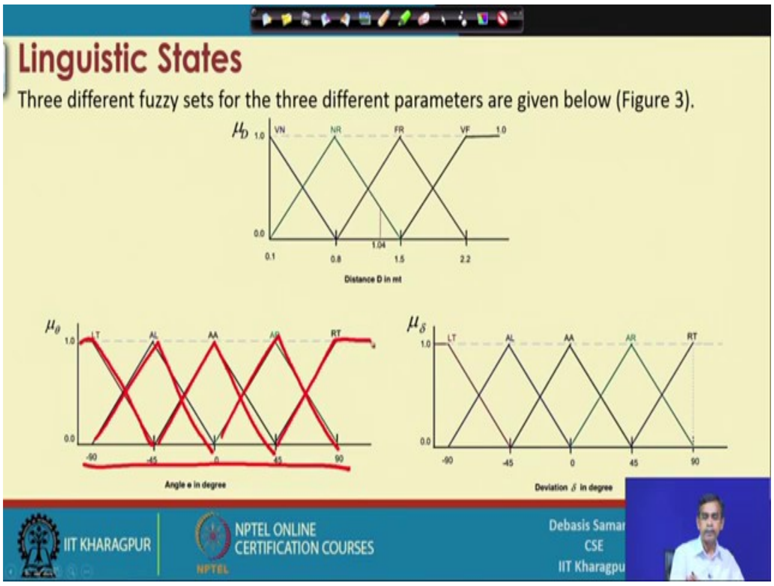


Now, similarly the angle that is the direction of the robots with respect direction of an object with respect to the robot or the deviation of the robot can be describes again in terms of five different linguistic states. That we have discussed about here. The five linguistic states are left, ahead left AL, ahead and ahead right AR and then right.

Now so, it is basically the prerogative of the fuzzy engineer who can disguise that about the particular input and then they can decide the fuzzy five states. So, we have discussed for four different fuzzy states for the distance and five different fuzzy sets for the angle. So, we can discuss alternatively three different fuzzy sets for distance also and then four different fuzzy sets for the angle or three different fuzzy sets angle or more than five different fuzzy sets also. So, it depends on the fuzzy engineer how he can plan it how can design it. So, it is totally depends on the expertise of the fuzzy engineer to decide the fuzzy linguistic sets.

And whatever the fuzzy states you decide it will work for you whether accurately or less accurately that is depends on the design actually. So, these are the, I mean two different inputs the D and θ the fuzzy sets can be defined. Similarly, for the δ also as it is an angle. So, the same fuzzy linguistic can be considered. So, for the θ the angular direction and for the deviation δ the same fuzzy linguistic can be considered here. Now, after having this one, we will see exactly the rule base. Now, we will consider, we will be able to discuss about the rule base.

(Refer Slide Time: 18:43)



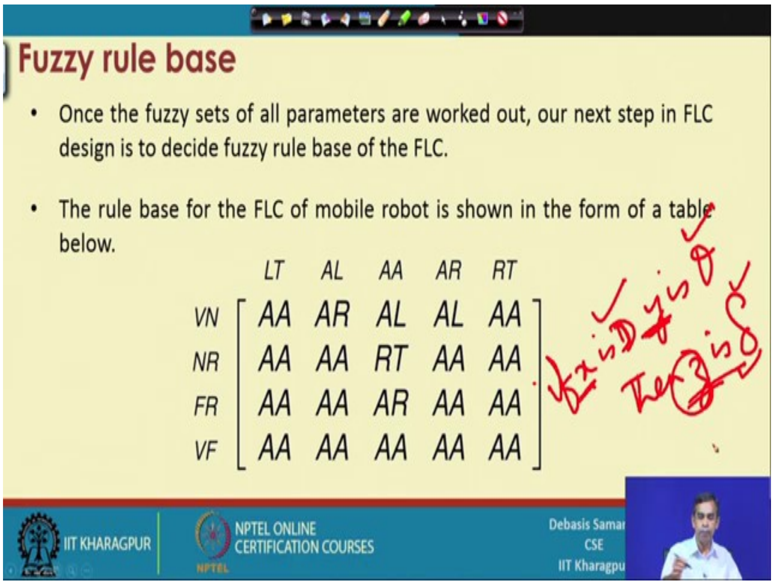
Now, before going to this for each fuzzy set they should be defined by their corresponding membership function. Now, we will discuss about the membership angle we know that the fuzzy membership function can be of either triangular shape or trapezoidal shape or some bale shape like this one.

So, in this example we consider the different fuzzy membership function for the different fuzzy sets namely, say distance related fuzzy sets, fuzzy linguistic like very near, near, far, very far using some triangular membership function. For very near the membership function is like this. Similarly, for the near the membership function is like this and then for far the membership function is like this and for very far the membership function is like this. So, we can easily understand that whether near object is like this, so membership function will vary like this one. So, this has certain meaning with respect to in our fuzzy uncertainty and then the corresponding the fuzzy elements there.

Now, likewise the membership function for the angle θ and then division also can be described. So, as we told you that the value ranges from -9 to 90. So, it is basically the range of values that is for the membership function should be and then for the different fuzzy linguistic like left it is defined here then ahead left, it is ahead, ahead right and then right this one. So, this way the fuzzy linguistic and then corresponding membership function is well defined and this is the same thing this is defined for the deviation δ and we can note that for θ and δ, their membership functions are same. So, it is quite possible and because angle and then this then derivation they have the similar magnitude and then similar of interpretation.

Now, having this fuzzy linguistics and then fuzzy membership functions, we are in a position to decide about the rule base.

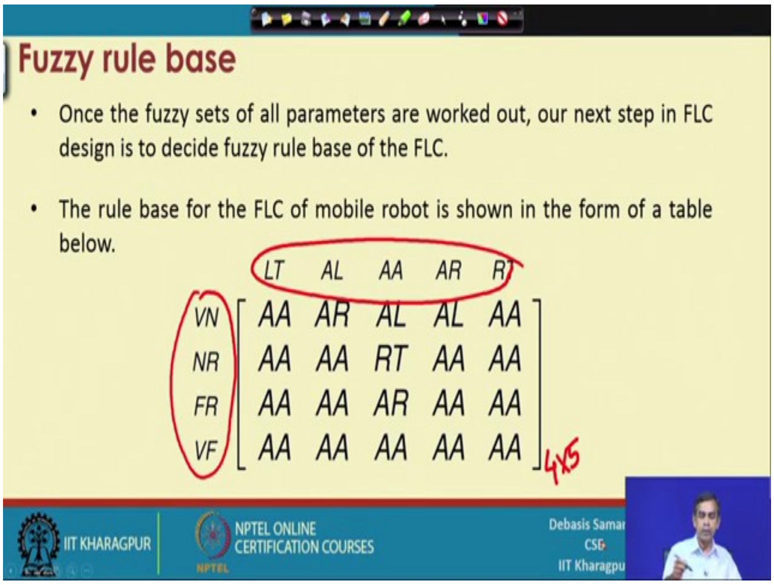
(Refer Slide Time: 20:52)



This rule base can be for simplicity, we can define this rule based in terms of a rule matrix and here basically the rule base signifies few things that for any particular value that belongs to the distance and for any particular value of θ, how the rule needs to be decided. For example, here one idea is that. So, the rule base it is like this as you know rule is taking this form **if x is D, y is** θ **then z is** δ, it is like this, this is a rule like this; that means, if the distance belongs to this one in a fuzzy linguistic and then there a rotation angular direction y is in terms of fuzzy linguistic then what will be the output in terms of fuzzy linguistic δ.

So, these are the fuzzy linguistic as you have already studied and these are the different input at any moment and then so for this different input how the output z can be obtained. So, this is basically objective and such things can be expressed using some rule matrix. So, these are rule matrix that is there and in this rule matrix all the fuzzy linguistic state related to the distance and then corresponding linguistics they relate to the angular direction is row and then column wise specified.

(Refer Slide Time: 22:06)

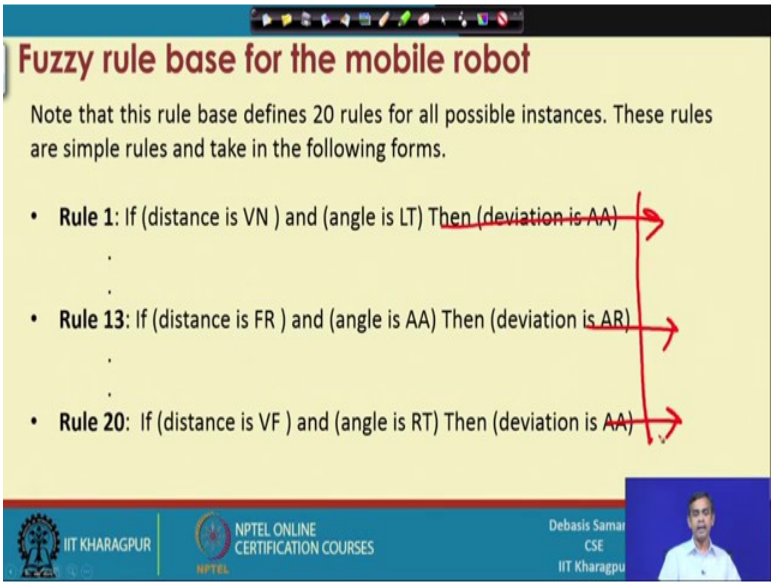


So, here if x is VN and then y AA, it says that see x is VN and y is AA. That means, the angle is ahead then rule this that direction δ is AL that mean deviation will be AL. So, it is like this. For example, if this is the one another output; that means, if it is FR and this is the angle direction so that direction will be AR. So, this basically says that different rules that can be fireable they that can be related to the fuzzy movement and this is expressed in terms of rule base.

And here so far the four linguistic sets are there which belongs to the distance input and then five linguistics are there, so for the input angular direction. So, altogether, the total number of rules that is here feasible these are ¿4×5 means 20 rules. So, in this fuzzy system and this basically gives the rule base. This is the rule base which is shown in the form of a matrix and we can shown the same rule base in the form of a fuzzy proposition that we have already discussed there. That means, **if x is D, y is** θ **then z** is this one these are the rule form look like, anyway.

So, we will discuss about these the rule base and you will see exactly how such a rule base can be used and then corresponding inference engine can be developed that is our next target.

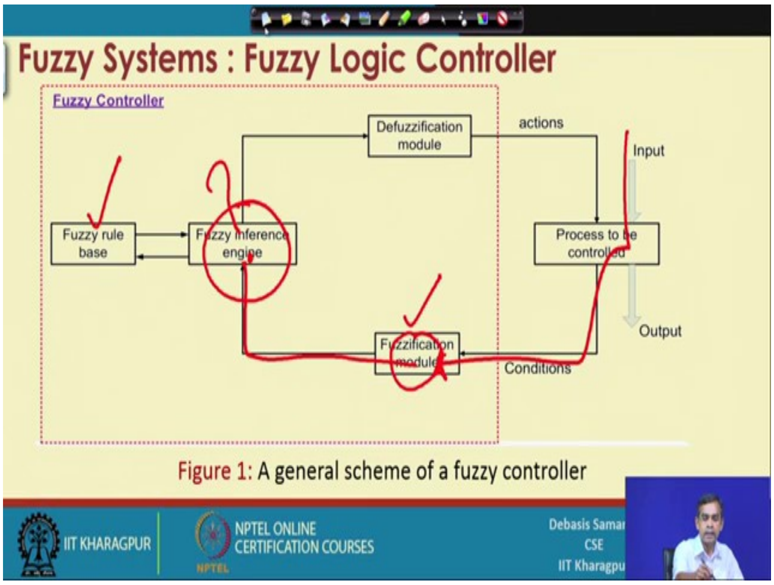
(Refer Slide Time: 23:47)



Now, so far the fuzzy rule base for the mobile robot is concerned I have told you. So, altogether there are twenty rules rule one is like this. So, if distance is VN and then angle is LT then derivation is AA. Similarly the other rules are there. So, all rules can be expressed this one and you know such a rules can also be expressed in terms of a rule and what is called a matrix relation matrix sort of thing. We have already learned that how such a rule can be stored in for matrix and all these metrics. So, then we can take all the rules and then corresponding all matrixes and then we can infer something from there that is the rule inference, whatever the idea we have discussed earlier they can be applied here.

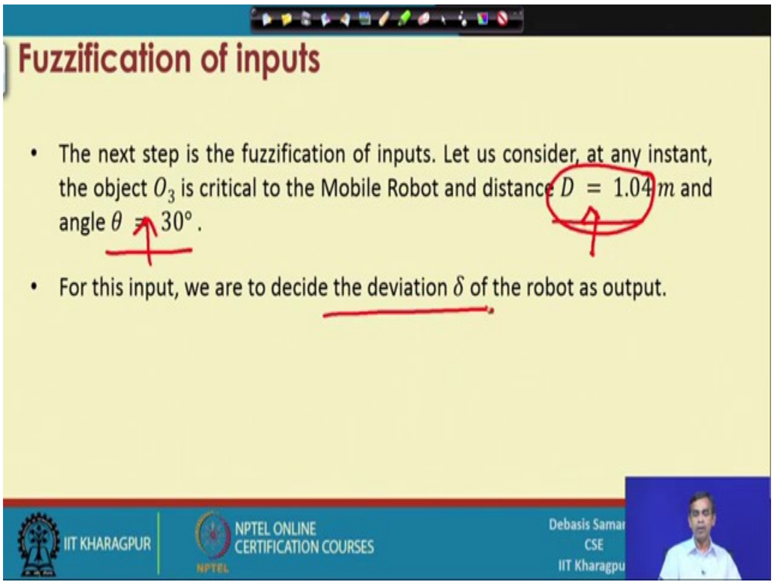
Now, in case of Mamdani approach they follow a little bit different idea rather more simplified and then sophisticated idea that is considered here. Now, see here the rule that, so we have learned about the fuzzy rule base just now, we learn about fuzzy; we have learned about fuzzy rule base we have learned about it.

(Refer Slide Time: 24:53)



And then we are we are in the process of learning the fuzzy inference engine and before going to fuzzy inference engine basically we have to learn about fuzzification module. So, our next task is basically what will be the fuzzification module is there. Now, fuzzification module basically takes some input and then this input is to go to the fuzzification module, take the fuzzified value and this fuzzified value will be used by the fuzzy rule engine. So, we will not be able to discuss the fuzzy inference engine until we discuss the fuzzification module and so we will discuss about the fuzzification module first.

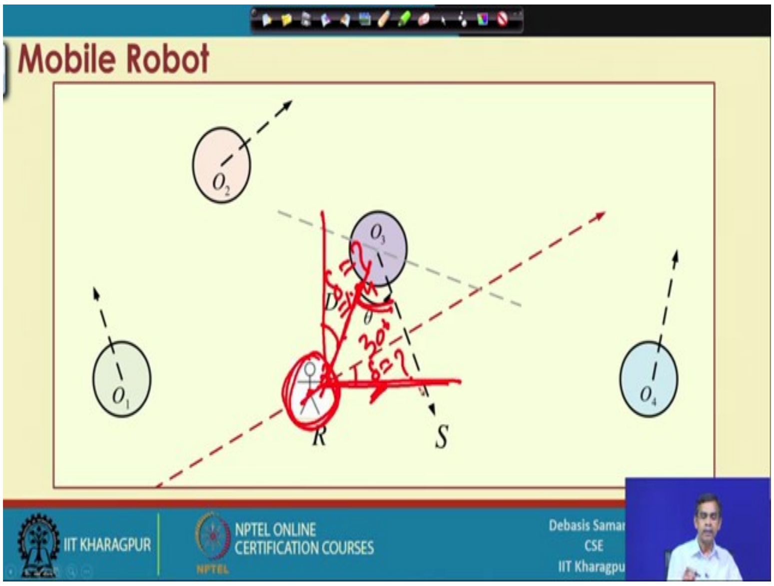
(Refer Slide Time: 25:40)



So, for the fuzzification of input is concerned we have to consider a specific instance for a particular values of the input. We will consider that at any instant a D that is the distance of the distance of an object from the robot is basically 1.04 meter and obviously, it will be in the range 0.1 to 2.2 that we have already specified there and at that instant also we assume that the angular direction θ equals to 30 degree. So, this is a specific instance and at the specific instance we will see exactly how the these are the crisp value crisp value of the input can be fuzzified in terms of whatever the fuzzy linguistic state that we have discussed their and then corresponding the fuzzy output and then finally, we will calculate that deviation δ of the robots.

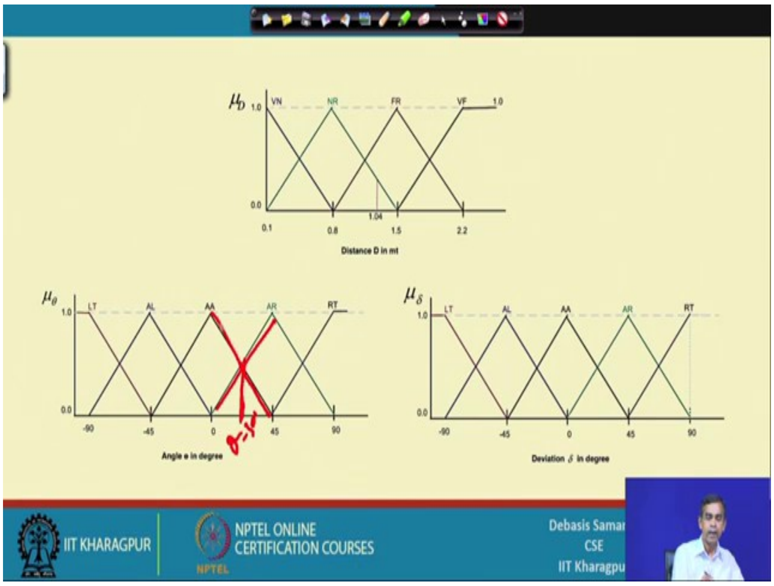
So, typically it is basically idea it is like this, so a particular example that I have already told you.

(Refer Slide Time: 26:38)



So, it is basically D these is the D and in this case 1.04 and this is a θ at any moment this is 30 degree. Now, here, this is the input that is available to the fuzzy logic controller of the robots and a taking this input the controller will dictate the robots that in which direction either with certain δ here or in this direction δ here that we have to decide. So, this is the likely value that singular value. So, fuzzy logic controller will take this input this D and then θ and then calculate δ and then it automatically the machine that is our tools that is there in the robot it will take this value and then direct the movement according to this direction or that one. So, this is the idea about it. So, our next task is basically how to calculate the fuzzy input for a given crisp input and, so it is here.

(Refer Slide Time: 27:43)

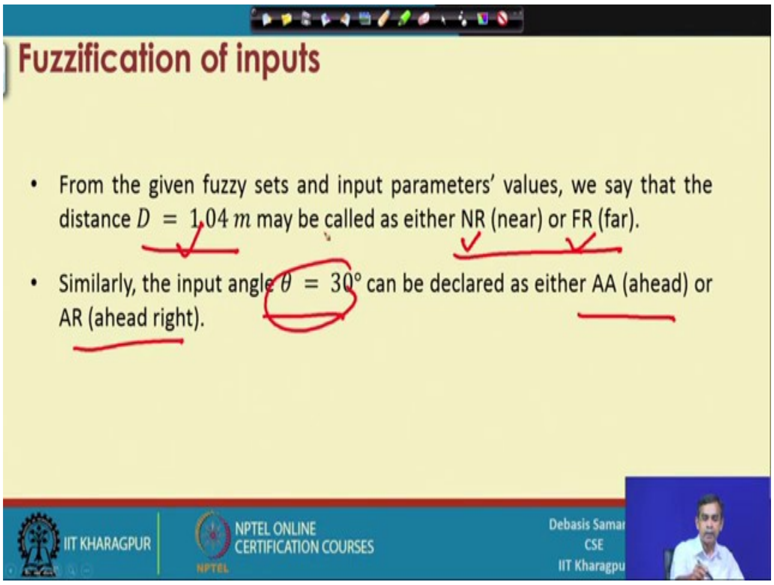


Now, that typically so far the is concerned here we know that these are the linguistic variables linguistic fuzzy linguistic for the distance and D at the moment is 1.04. So that means, this is the crisp and then corresponding crisp the fuzzy output is it fireable to NR that means so far the near is concerned it is that this value with these a membership function.

Again for the same this is the also fireable to the fuzzy linguistic far; that means, D being the distance 1.04 it belongs near fuzzy set as with this membership value and it also belongs to the FR fuzzy set with this membership value. So, these are the fuzzy input actually you will calculate it and then we will use it. Likewise for θ equals to 30 degree, these are the, both ahead and AR ahead right are applicable to this so far the fuzzy input is concerned. So, if this is the θ equals to 30 degree then curves on θ the fuzzy input will be AA and AR.

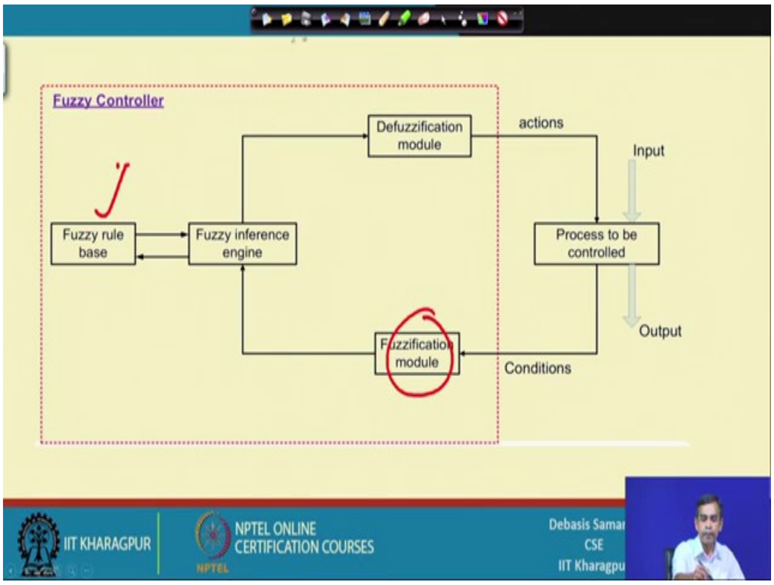
Now, in the next discussion will discuss about how; in the next we will discuss about how such the procedure general procedure for obtaining for a crisp input the fuzzy output, fuzzy input.

(Refer Slide Time: 29:16)



So, in this particular context of example as we have learned it, D 1.04 if it is a fuzzy crisp input then it has the fuzzy output near and FR, but near with certain membership value and FR with certain membership value. We will be able to easily calculate and we will see the calculation in the next lecture how the near fuzzy sets and FR fuzzy sets for this input can be obtained. That is basically called the fuzzy input or fuzzification for the input. And likewise for θ 30 degree which graphically I appear as ahead and ahead right both are feasible and then what is the corresponding membership value. Therefore, the fuzzy sets that for this input can be obtained. So, all these things will be discussed. So, that will be discussed in our next lecture.

(Refer Slide Time: 30:07)



And then, so till time we have discussed about the fuzzy rule base fuzzy rule base design and then we will discuss about the fuzzification module, once we learn it then go to the fuzzy inference engine which will be covered in the next lecture.

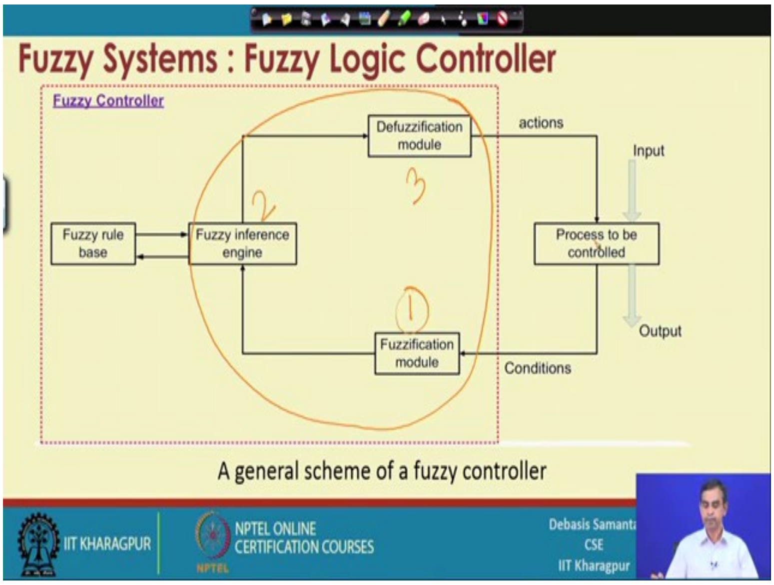
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture 12
Fuzzy logic controller (Contd.)

So, we are discussing the design of a fuzzy logic controller. In the last lecture, we have learnt how the rule base which is an essential part of a fuzzy system has been developed. So, will discuss about the other, there are other modules. So, today we will discuss other module namely these three modules. So, fuzzy inference engine, fuzzification module and defuzzification module.

(Refer Slide Time: 00:32)

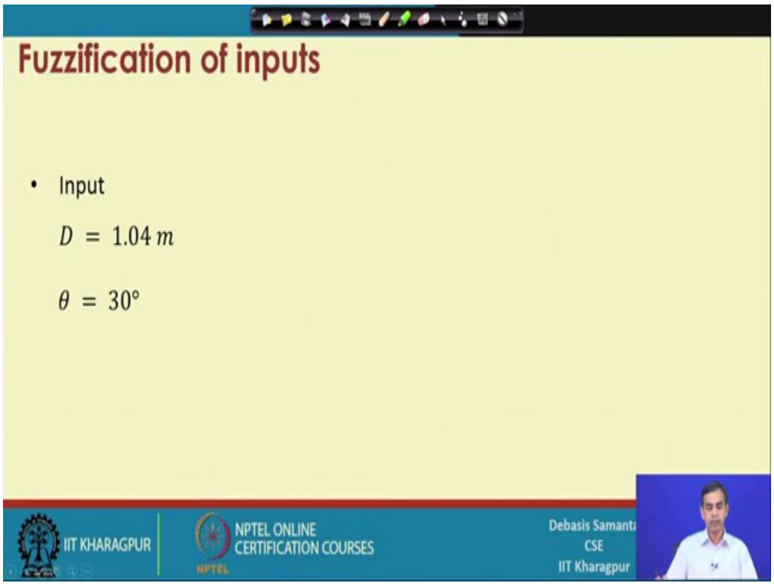


Now, so first will discuss the fuzzification module because the input it will take the data that is required to control the system and the fuzzification module will produce an output which will be used by the fuzzy inference engine. So, we should learn fuzzification module first then we will be able to discuss about the fuzzy inference engine.

Fuzzy inference engine will consult with the fuzzy rule base and then produce an output that is the fuzzy output. So, fuzzy output will be the input to the defuzzification module that is the final module in the fuzzy logic controller and we will discussed in the fuzzy logic defuzzification module. Defuzzification module will gives the output that output will be used as a crisp value and then helpful for the controlling some application.

So, this is the task that we are going to learn today fuzzification module followed by the fuzzy inference engine and then defuzzification module and will follow the Mamdani approach because we are discussing Mamdani approach first.

(Refer Slide Time: 02:03)

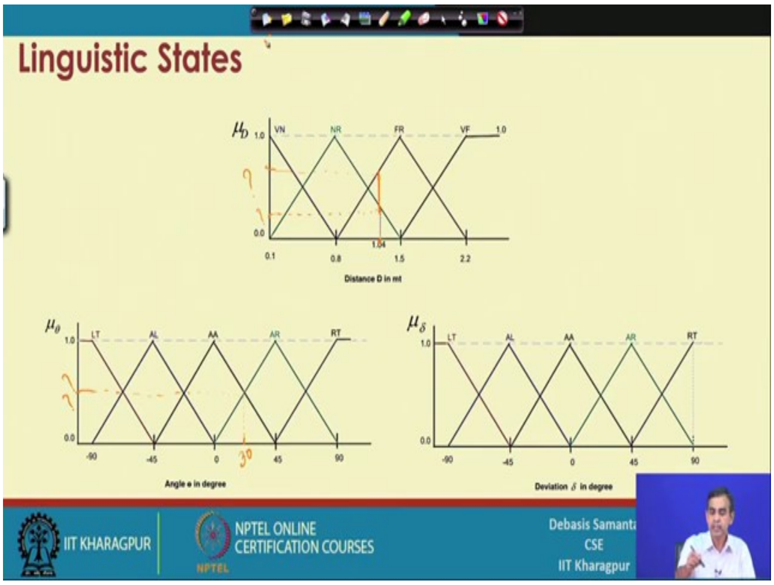


Now first will discuss about the fuzzification input. So, fuzzification input as we have already discussed that for a given input that is usually the crisp input and then how this crisp input can be converted to the fuzzy input.

Now, the system that we are discussing in these application is a mobile robot and we have already mentioned that mobile robot has the two input namely the distance and then the angle angular direction of a moving object. So, here let us consider for an example at any instant the distance from the robot to a moving object as an input and this is the input and angular direction. That means, in which angle making an object moving towards the robot and let it be θ and θ is value at the current time instant is 30 ° .

So, these are the two input then these are the two input will be given to the controller as a crisp input. So, fuzzy controller will transfer this input into the fuzzy output. So, we will discuss how the two input, these two input can be converted to the fuzzy input.

(Refer Slide Time: 03:27)



Now, in order to understand how these fuzzy how the crisp input can we convert the fuzzy input we can take the care about the membership function for each fuzzy linguistic state that we have already discussed. So, far the fuzzification of the system is concerned and there we can recall we have discussed two and three membership function rather two for the input and one for the output for the input we have discussed about that distance as a three fuzzy linguistic four fuzzy linguistic namely very near, near far and very far. And similarly for the angular direction, we have considered five fuzzy linguistic left, ahead left, ahead, ahead right and then right. So, we have to take the fuzzy input in terms of these fuzzy linguistics.

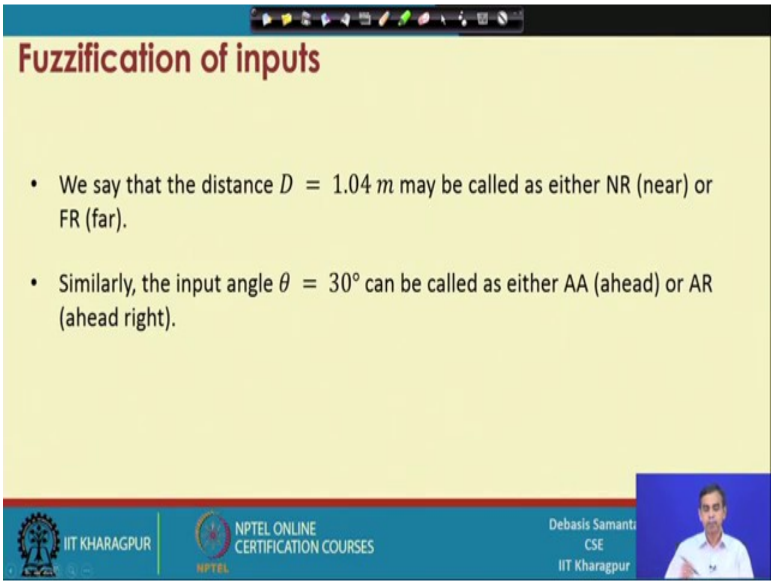
Now, for an example here as you have considered D is 1.04 this is the crisp input. So, so for the distance is concerned and so for the fuzzification is concerned, D 1.04 is this is the element and this element has the two fuzzy states namely near and far because it cover both the things. So, this means that that crisp input 1.04 as the distance can be considered fuzzily in terms of two fuzzy values the fuzzy near and then fuzzy far. So, distance 1.04 has the fuzzy membership belongs to the fuzzy state NR and 1.04 also is a fuzzy member belongs to the fuzzy state FR, but they belongs to the two fuzzy sates NR FR with certain degree of membership values. So, we have to calculate the degree of membership values for these two elements which belongs to NR and FR.

Now, first 1.04 being the distance and if it belongs to fuzzy state NR then the membership value can be decided by these. That means, this is the membership value is this one. So, we have to calculate what is the value this one. Similarly if you consider the FR fuzzy state distance also belongs to the FR fuzzy state then. So, these membership value this one. So, 1.04 crisp input belongs to two fuzzy state NR and FR having the two different membership values for NR this one and for FR this one and you have to calculate these two values first.

Likewise, angular rotation angular rotation that we have considered in this example that θ is 30 ° . So, θ is 30 ° mean this one 30 ° . 30 ° being the crisp input has the two fuzzy states, one is ahead and then ahead right. So, if it belongs to ahead then the membership value can be computed this one. So, this is the value that we have to know. Similarly if it belongs to ahead right then it has a membership value the same here in this particular case only. So, these two membership value needs to be calculated and then finally, output will be considered that will be discussed later on.

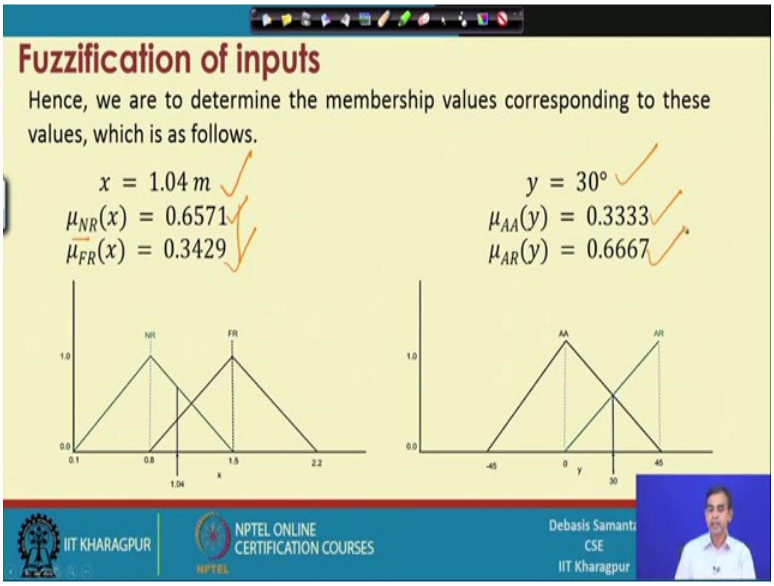
Now, let us see how the input, but input values the fuzzy values with their membership can be calculated. So, this can be calculated, this can be calculated for example, fine.

(Refer Slide Time: 07:12)



So, we can understand that the D distance 1.04 can be called either NR or FR; that means, they belongs to the two fuzzy state with different membership values. Similarly, the angular orientation θ 30 ° can be called as either ahead or ahead right with the different membership value and we are in the process of calculating the membership values of the their there.

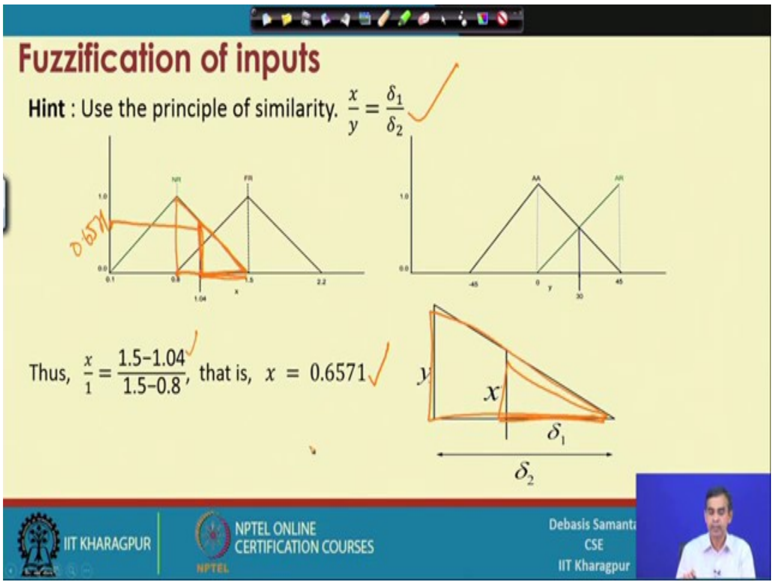
(Refer Slide Time: 07:35)



So, here is an example how the membership values for the two inputs one is distance and another is angular orientation can be calculated. So, it is a calculation, calculation shows that the membership values for the distance $x=1.04$ which belongs to the fuzzy state NR can be calculated as this one. Similarly, the membership values for the input x which belongs to the fuzzy state FR can be calculated this one. Likewise, the membership values for the other input $y=30°$ which belongs to the fuzzy state AA can be calculated as like this and AR can be calculated like this. Now, question that is how this calculation obtained.

So, this calculation can be obtained using similarity of triangle and it is very straightforward calculation. So, let us see how this calculation can be calculated here.

(Refer Slide Time: 08:38)

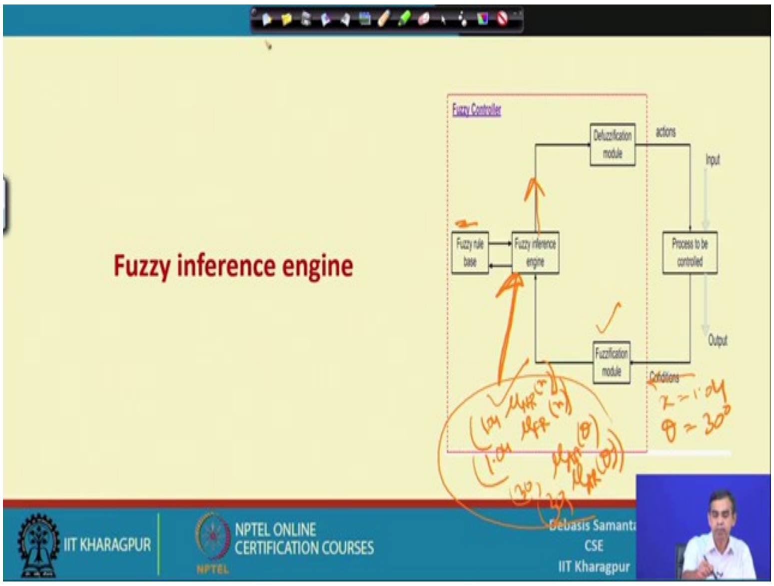


So, idea of this calculation is given like this. So, it basically follows the principle of similarity. Now principle of similarity means if you consider, this is the one triangle and this is another triangle. So, these two triangles are similar triangle if they are similar then we can write $\frac{x}{y}=\frac{\delta_1}{\delta_2}$, if this is the δ_1 and if δ_2. So, it is a formula that we have from the principle similarity of two triangles, we can follow this one. Now, the same logic can be applied to calculate the different membership values. For example, here if you consider distance 1.04 as the input and we want to calculate, this membership value NR right. So, we can consider this is the one triangle and the entire triangle they are similar triangle.

Now, if we considered this things then for this it is basically these values $\frac{1.5\text{-}1.04}{1.5\text{-}0.8}$ and then if we considered this is basically x and this is the 1. So, it $\frac{x}{1}=\frac{1.5\text{-}1.04}{1.5\text{-}0.8}$ this one. So, this way x will be calculated this one. So, x means it is basically this is the x it is calculated this one. So, this value is basically 0.6571 in this case. Similarly, for the angular orientation θ we can calculate likewise. So, it can be calculated like this. Now fine, I just forgot one thing more. So, this is for near. Similarly for FR also we can calculate. So, if we calculate the FR then we should take this is the triangle similar to this is another triangle. So, using these two triangle similarity, we will be calculate this value. So, these value can be calculated the 0.333 that we have already learned about it.

So, this is the way that the value can be calculated and the same approach can be extended to calculate the membership values for other input $\theta = 30°$. We can follow this is the one similar triangle and this is one similar triangle to calculate this is the value and if we considered for the AR we have to considered this similar triangle then this similar triangle, so that we can calculate this one. So, both we can be calculated and the result can be obtained. So, the result that can be obtained is shown here, so the result that can be obtained that can be obtain easily and then can be used for the next what is called the next step.

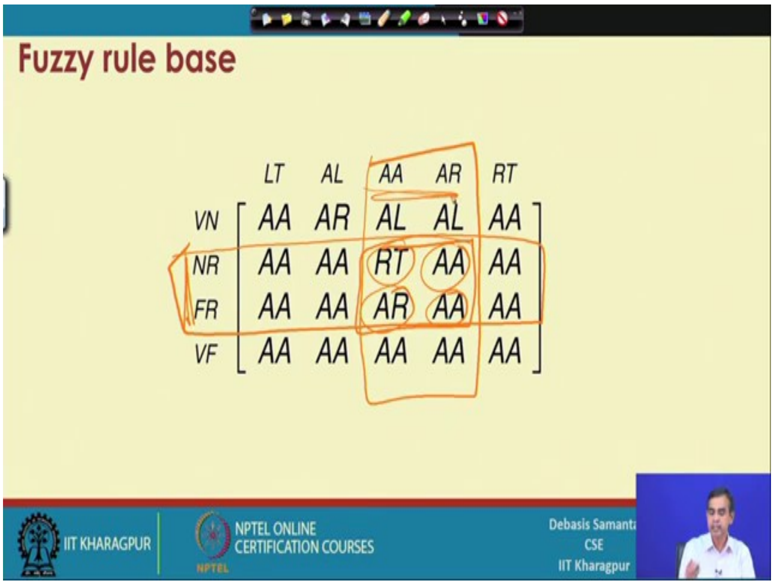
(Refer Slide Time: 11:30)



Now, our next step, what we have learnt is the fuzzification module given an input. So, for $x = 1.04$ as the crisp and $\theta = 30°$ the two input and fuzzification module will give you the different $\mu_{NR}(x)$, $\mu_{FR}(x)$. Similarly, $\mu_{AA}(\theta)$ and then $\mu_{AR}(\theta)$. So, these are the value; that means, 1.04 for read this is the one fuzzy element, 1.04 is the fuzzy state and similarly $30°$ is a fuzzy state having the membership value $30°$ is the fuzzy state having the membership value. So, all these values will be used as an input to the fuzzy inference engine. So, these value will be used to the fuzzy inference engine and then fuzzy inference will use these values and consult the fuzzy rule based and then produce an output this is called the fuzzy output.

Now, will discuss how the fuzzy inference engine take care these values and consult the fuzzy rule based and produce a fuzzy output that is basically the idea about fuzzy inference engine and will discuss how the fuzzy inference engine can be implemented using Mamdani approach.

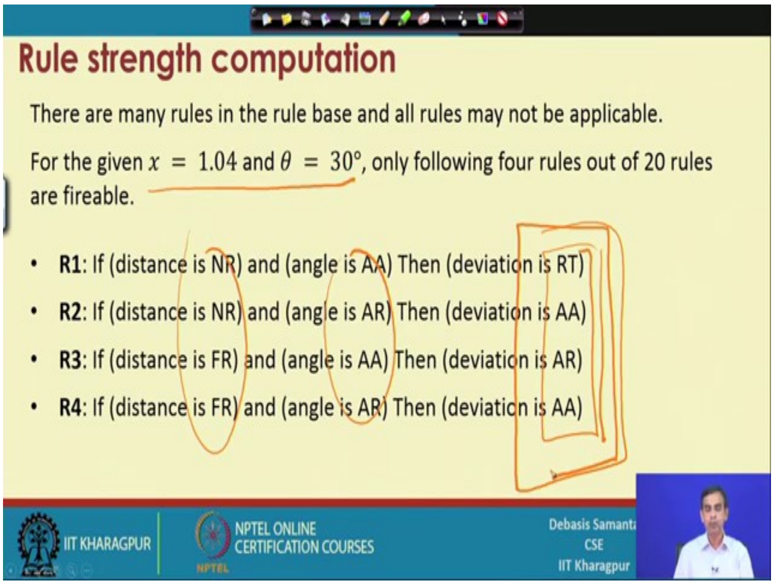
(Refer Slide Time: 13:09)



So, the idea is that fuzzy rule base is here that we have already learned and in our fuzzy rule base in the context of current mobile report example we know that 20 rules are there. All the rules are depicted here in the form of a rule matrix. So, out of these 20 rules, we have to decide particularly fuzzy inference engine will decide which are the rules are exactly useful, so far the current input is concerned. That means, out of 20 rules all rules may not be applicable in the current context. So, fuzzy inference engine will take a calculation which basically try to see out of these 20 rules, how many subset of rules those are basically related and can be considered to calculate the fuzzy output.

So, that is the objective of this fuzzy inference engine. So, fuzzy rules those are the 20 rules we have discussed about it that can be considered here.

(Refer Slide Time: 14:08)

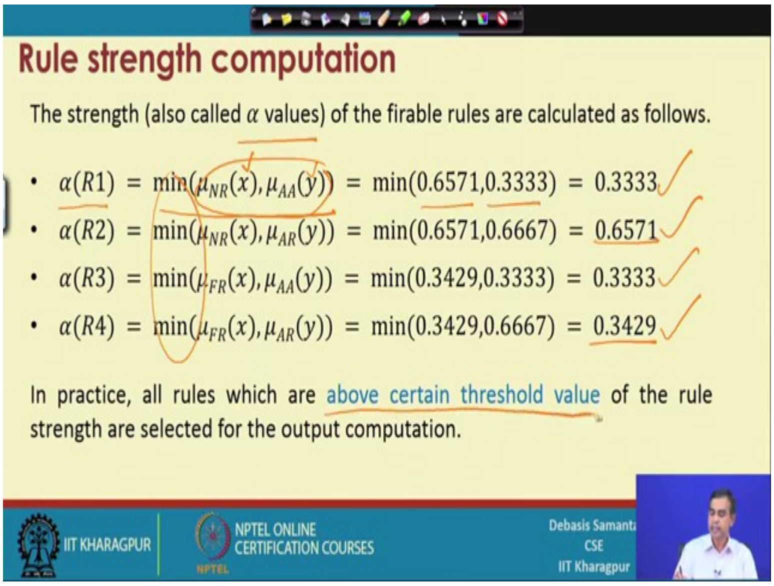


Now, one thing you can understand here that in the current context again we see out of this four linguistic only near and far are the important. Similarly out of the five linguistic states related to angular orientation only these are important. Now if you consider then, so then intersection of these things means only these four rules are basically important. That means, *if x is NR and y is AA then deviation is right* one rule *if x is NR and y is AR then deviation is AA* another rule, *if x is FR and then y is AA then the angular deviation is* θ *AR* and *if x is a FR and then y is AR* this one. So, out of 20 rules in this particular context depending on the states so far the distance and then angular orientation is concerned only four rules are irrelevant. So, these four rules we can display here list here. So, these are the four rules that we have just know identified if distance is NR and angle is AA then deviation is RT and so on. So, these are the basically four rules that is relevant in the context of current input.

So, at any instant fuzzy logic controller will receive this input fuzzify it, these are the fuzzified values and based on this fuzzify it basically decide what are the rules that can be favourable and then it will see that these are the most relevant or significant rule, so for the decision of output is concerned. So, fuzzy inference engine will take care about it take an input and then select the subset of rules which are relevant and then it does more in fact, it basically out of all the rules it can take care all the rules and then decide the output. But to be an accurate and then more efficient it will try to find some rules which is already selected or shortlisted here. So that it can be more accurate and that is why out of these four rules again it apply to rank them and then decide which are the rules are mores strong. So, it basically Mamdani approach is the next step in the inference engine needs to basically compute the rules strength of the selected rules.

Now, how the rules strength can be computed I have given an idea just I will like to give an idea about it.

(Refer Slide Time: 16:55)



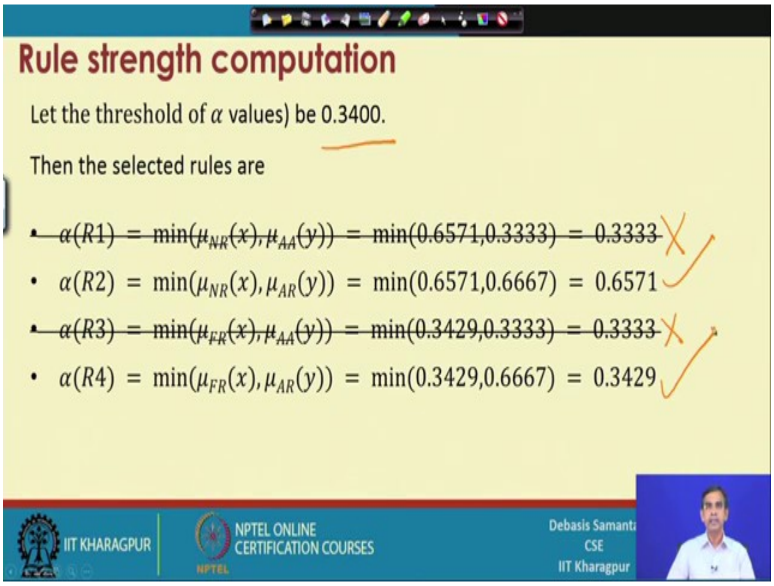
So, it basically calculate is very simple approach to calculate the strength of each rule and this is denoted by call α values. So, if *R1, R2, R3* and *R4* are the four rules the α value can be calculated like this it basically take the minimum of the fuzzy membership values for the input. Now, in the context of first rule the two membership values related the fuzzy state NR and the fuzzy state AA having *x* element are the input and then we have already known the membership value of this. So, it will take the minimum of the two membership values as the rules strength.

Now, in the current context the $\mu_{NR}(x)$ can be calculated as this one and $\mu_{AA}(y)$ also calculator 0.33. So, it taking the min, it gives the rules strength of the first rule likewise rules strength for the second rule can be calculated this one and this one and this one is taking the min of the different membership values belongs to the different states corresponding to particular input. Now, out of these four rules then its selects right some rules which are basically based on some threshold value.

Now, if we take the threshold value is 0.300 then all rules will be selected now if we select on the other hand the rule strength is 0.3400 then only these rule and these rule will be selected. So, it depends on the threshold value and the threshold value will be decided by the fuzzy engineer from his own experience or using trial and error method anyway. So, some threshold value is required, in order to select again from the shortlisted rules, so fuzzy engine fuzzy inference engine take a value of a threshold value and then based on this threshold value, it will compute the rules strength and then based on the threshold value from the rules strength it select the stronger rule which is basically above the threshold value.

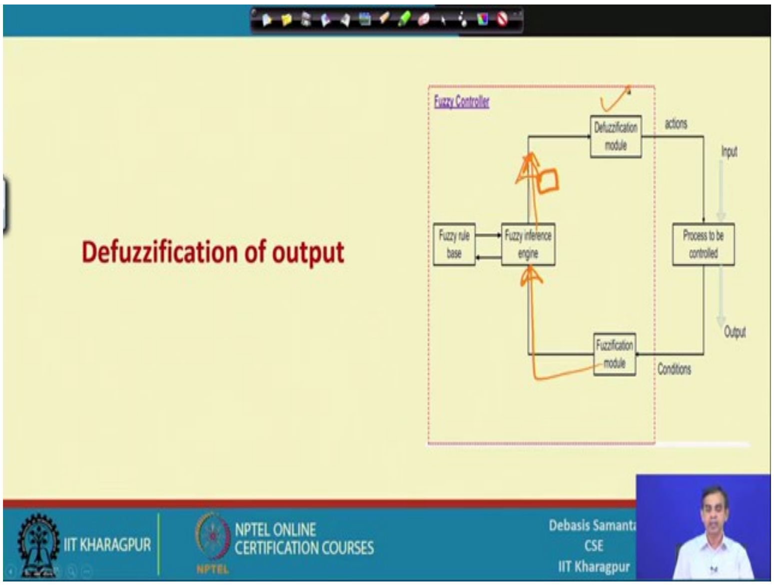
Now, if we consider for an example say, rule strength the threshold value is 0.3400 then the rule that will be selected shortlisted is there , if you consider threshold value is 0.3400 then this rule will be ignored, this rule will be ignored only this rule will be considered.

(Refer Slide Time: 19:21)



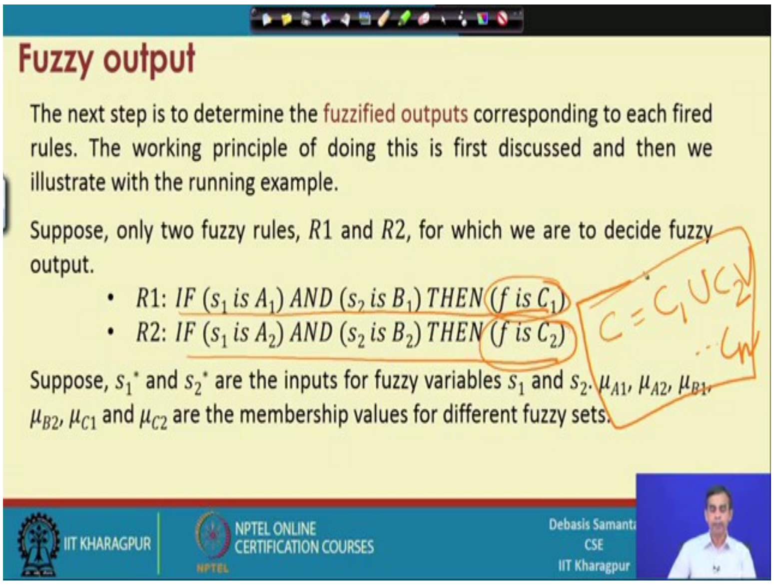
So, out of the twenty rules because of the relevancy we have selected four rules and then again by means of rule strength computation we have shortlisted from the list of selected rules two rules only. So, these two rules will be used to calculate the fuzzy output. So, this is the task up to the fuzzy inference engine fuzzy inference engine concern the rule based for a given input and then select the rules according the relevancy and from the relevant rules it again compute the rules strength and based on the rule strength computation and threshold value we select the final list, a final rules those are basically will be used to calculate the input no output of the fuzzy system.

(Refer Slide Time: 20:29)



Now, our next task, fuzzy inference engine basically take the input fuzzy input and then produce the selected rules these selected rules basically gives us the output. So, we can say the fuzzy inference return the selected rule, the strong the strong rules and then from this strong rules we can have the fuzzy output. Now, we will discussion about how the fuzzy output and corresponding defuzzification of this output is there. So, these the final stage of the fuzzy a logic controller defuzzification of the output and will discuss this fuzzification, defuzzification method in the next few slides.

(Refer Slide Time: 21:12)

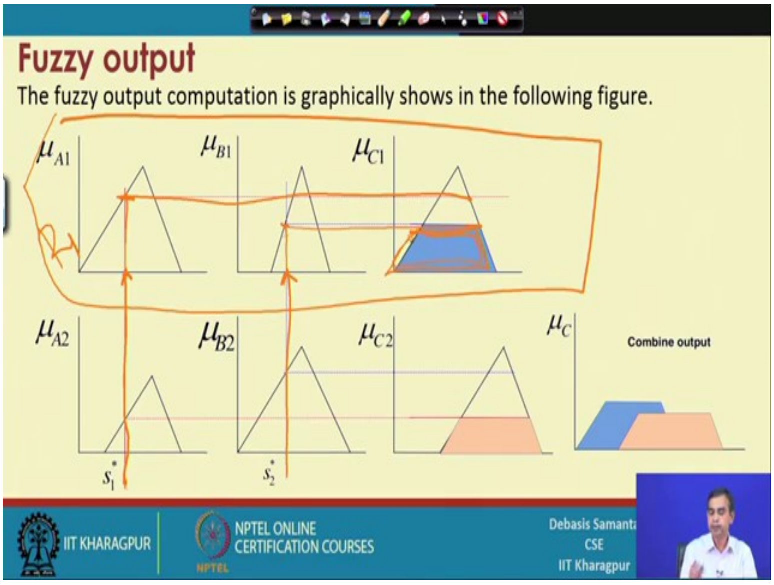


Now, the idea is that, that fuzzy inference engine returns some rules those are more appropriate. So, for the decision is concerned corresponding to some input. So, these rules are basically essentially the output for an example. So, suppose is just any two rules are like this. So, this is the one rule and these are two rules. So, output of this rule is basically C_1 that is the output and output of this rule is C_1, C_2 .

Now, if we combine the two outputs then the result and output C is basically union of the two output C_1 and C_2 . That is the idea that is followed in Mamdani approach there. If there are more rules other than two rules say they are N rules then it will take the union of n output that is there and so output for the system it is basically in a fuzzy way it is the fuzzy output. So, it is like this now in our current example. So, that there are two rules you have selected *if x is NR and y is AA, then δ is AA* that kind of things are we have considered. So, the output is there.

Now, we will see exactly how given such a rules the output can be calculated.

(Refer Slide Time: 22:48)

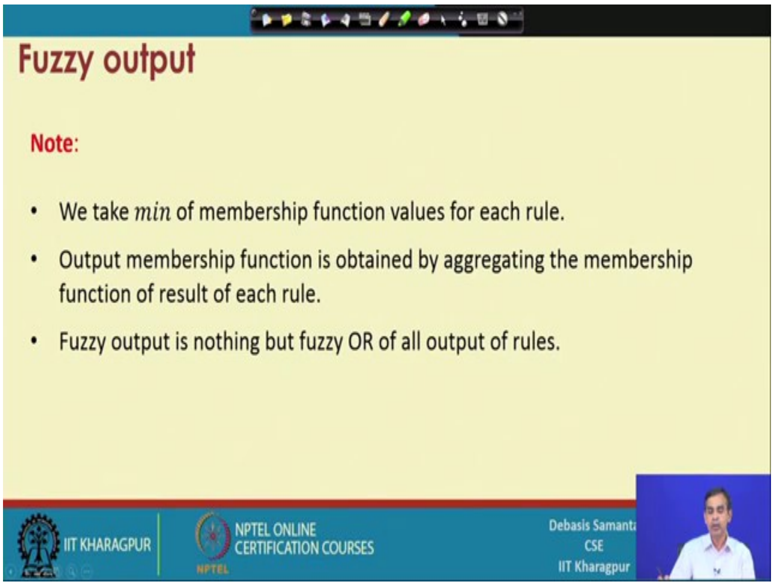


We will, although given example of graphical method, but the same thing can be done in a mathematical way which you have already discussed while we are discussing about defuzzification concept using centroid method or maximum method or weighted minimum method or whatever it is there. So, will consider, any method can be applied here. Now, let see how the output of a fuzzy system based on the different rule can be computed.

Now suppose, this is the rule 1 this is the rule 1 and then graphical representation of the rule one it basically says that if μ_{A1} for an input and if this is the fuzzy state and if μ_{B1}, for an input and this is the fuzzy state and this is basically the output of the fuzzy state. So, for particular values of input, it basically take, this is the basically input and for another input is there. So, two input are like S_1 and S_2 and then these are the basically rules there are this is the membership value for the first input and this is a membership for the first that we have already calculated. Now, if we draw a line joining this thing and parallel to the horizontal axis then it cut this one and if you like this and this one.

Now, these are the basically output, for the input is concerned, this is the output and. So, far the input is concerned this is output. So, Mamdani approach say that out of the two output you take the minimum of the two. So, these are the two output correspond the input this one and this one and we taking the minimum of this. So, this is a resultant output so the rule 1 is concerned. If this is the rule 1, for the rule one this is the output. Now, similarly for the rule 2 again these are the input corresponding the fuzzy state and these are another input for the another fuzzy state and if we take the minimum of these two then this is the minimum there. So, this is basically the output C_1 and these basically the output C_2 .

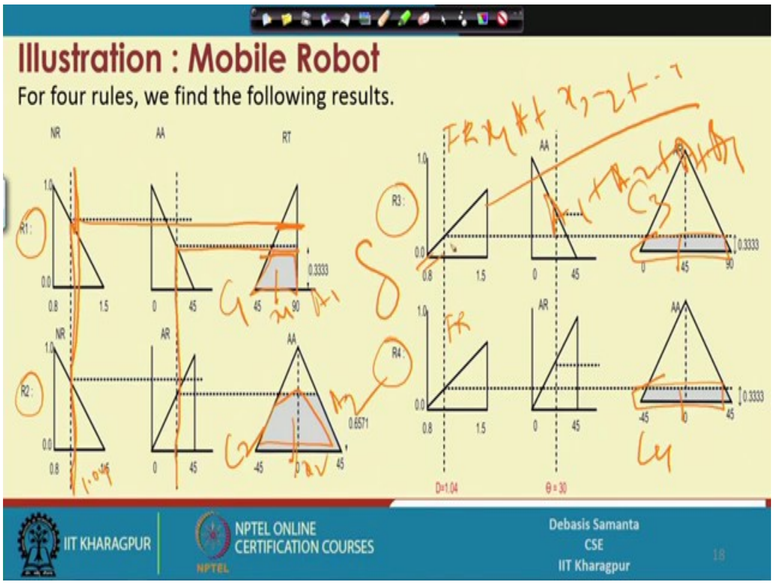
(Refer Slide Time: 24:47)



So, if we plot both the output on to the same graph the resultant, resultant output will look like this which is shown here. So, this is the resultant output.

Now, that defuzzification of these output can be obtained either reasons COG method or COS method and we will be able to obtain the crisp output from this one. So, now, let see how in the context of mobile robot we can have the data here. So, they are will consider the four rules suppose right in the last example that we have consider only two rules I will consider of four rules that we have so shortlisted there.

(Refer Slide Time: 25:31)



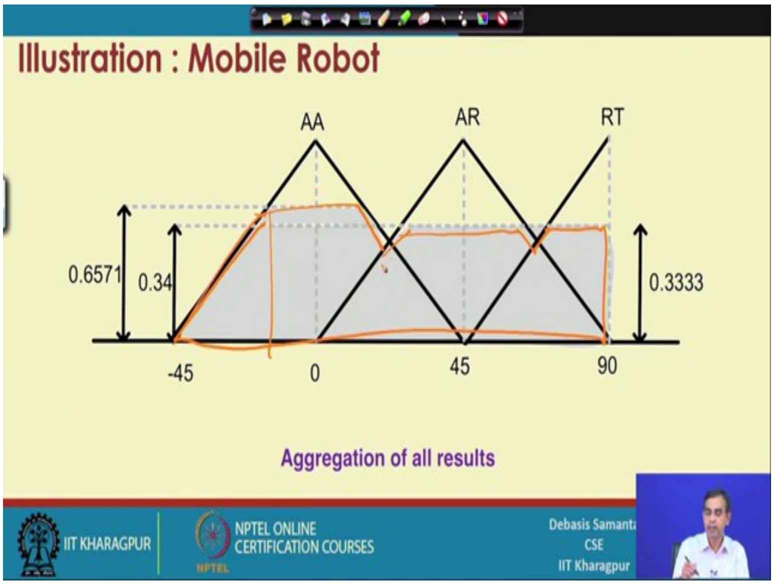
And then how the 4 rules for an example two rules can be considered then it will be most simple I want to give an example with the four rules that have been shortlisted based on the relevancy *R1, R2, R3* and *R4*. Now, let see how the output can be calculated now here this is the basically input. So, far the distance is concerned and this is the input. So, far the angular orientation is concerned and this is the output regarding the orientation here RT and AA.

Now, if this is the distance *x* and it will cut the value there for this is for the NR and this is this is for NR for the rule 1 and this is for the NR for the rule 2 and this is basically distance on 0.04 for the rule 3 and this is distance NR for the rule 4. So, it is basically NR and this is basically FR, this graph is for FR, this is for FR and this is for FR and this is the angle of orientation like this. Now again, rule 1 if we fair with $x = 1.04$ right it will basically cut this one and take the value. So, this is the membership output that is for the rule. Now, if you take another input θ then it basically cuts here and this one. So, taking the minimum, this is basically the value of the output for the rule *R1*.

Similarly, value of the output for the *R2* can be obtained as this one. So, so for the rule 3 is concerned value of the output will be this one. So, for the rule 4 is concerned value of the output will be this one. So, for four rules we got four output C_1, C_2, C_3 and C_4. Now from this 4 output we can calculate the fuzzy. So, from this four output fuzzy output we can calculate the crisp output value that crisp output value can be calculated using COG method or COS method. If you follow COS method for example, we take this area and the middle value, take this area and middle value, take this area and middle value take this area and middle value, and use the equation that $x_1 \times A_1$, if it is x_1 it is A_1 the area, if it is x_2 and A_2 and x_3, A_3 and x_4, A_4 and then formula is $\frac{x_1 A_1 + x_2 A_2 + \ldots}{A_1 + A_2 + A_3 + A_4}$. So, this will give you the output so, δ value. This way it can be calculated.

Now, let see what is the results that we can have based on the calculations COG the method. In case of COG method all the output can be plotted on the same graph, this, this can be plotted on the same graph and then from the resultant graph we can apply the COG method calculation so that the fuzzy output can be calculated.

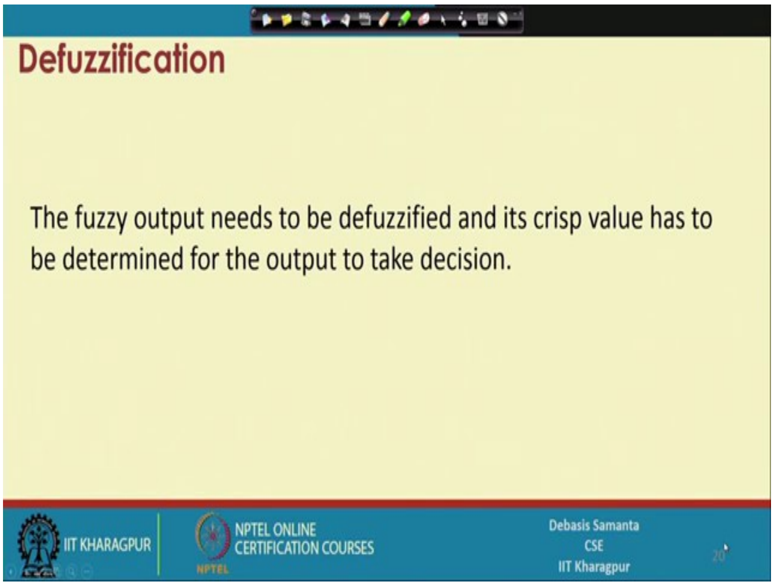
(Refer Slide Time: 28:47)



Now, I can give an example here for example, if we plot the four different output into the same graph the graph will look like this. So, it is basically this is the graph than this graph then this, then this, then this, then this. So, this is the output graph actually for the four rules related to the input *x* and θ. $x = 1.04$ and $\theta = 30°$.

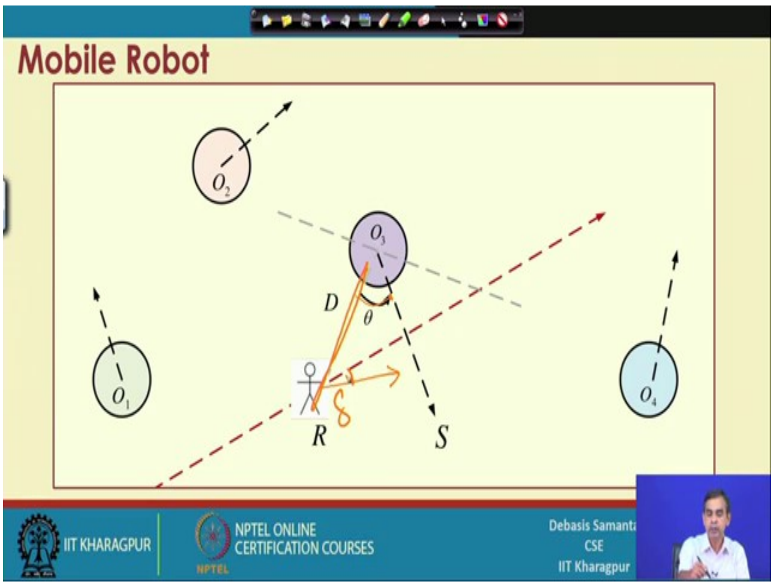
Now, the value can be calculated using COG method is a segment and whatever the method it is there will be able to calculate it that calculator is little bit combustion here. So, if it is the calculation is too much difficult for the COG the method then we can follow COS method or some weighted minimum method whatever it is there now using the COG method for the same thing the result that can be calculated is shown here and that is basically the defuzzification.

(Refer Slide Time: 29:34)



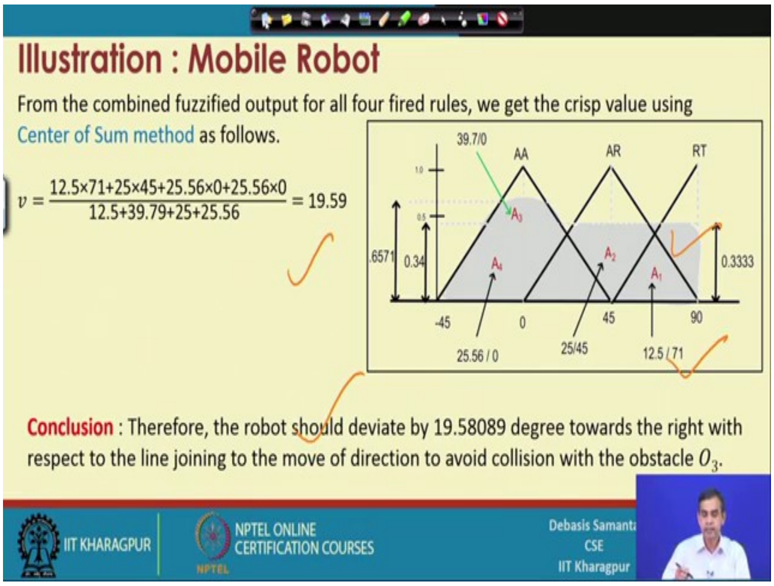
So, fuzzy outputs needs to be defuzzified and the crisp value has to be determined for the output to be taken. In this current context in the current context, this is the input here this is the input and this is another input and then output is basically δ regarding that in which direction it should move.

(Refer Slide Time: 29:44)



So, that basically obtained by the fuzzy output and then corresponding the crisp value if we follow it can be obtained the result.

(Refer Slide Time: 30:10)



Now, so far the current example is concerned the result that can be obtained using some COG method here we have apply the COG method like and it can be calculated as this one 19.59 or $20°$. So, this means that the robot that has at the current instance seeing an object this is the seeing and object a current instance seeing an object right it will basically move to the right a positive because this is a positive value to the right as a angle with respect to 3, the 30 ° 20 ° that is the angle. So, that fuzzy logic controller will take a decision about that. It has to do it from it state path towards the right by 20 °.

So, this output then can be given to the process controller and process controller will take this input and then accordingly controller will move or change the path of the robot. So, this is the one example that we have discussed about it. So, for the fuzzy logic controller is concerned and we will discuss next, another logic controller design that is Takagi Sugeno approach and in our next lecture we will follow that Takagi Sugeno approach in this regard.

Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 13
Fuzzy logic controller (Contd.)

So, there are 2 broad approaches, so far the fuzzy logic controller design is concerned. One approach is Mamdani approach and another is Takagi Sugeno's approach. Two approaches are different the way they treated the fuzzy logic controller design. We have discussed the Mamdani approach and we see that the Mamdani approach follow a rule base and then fuzzy linguistic state and then Fuzzification of the input and then produce the fuzzy output and then crisp value. The method more or less same, they are in the Takagi Sugeno's approach, but the though the way they treat the fuzzy inference engine is different.

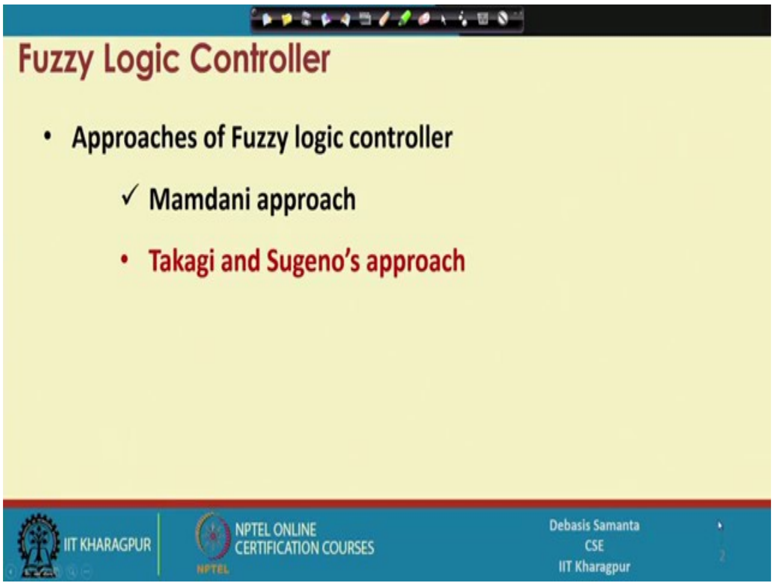
Now, fuzzy inference engine rather more what is called the interpretable so far the Mamdani approach is concerned. However, in case of Takagi Sugeno's approach, it is less interpretable. So, interpretation means, anybody can see from the design how it works, but Takagi Sugeno's approach is bit difficult because it follows certain mathematical treatment so far the inference engine is concerned.

Now, so far the output quality is concerned, the Mamdani approach is less accurate, whereas Takagi Sugeno's give more accurate output and output calculation if we consider according the to the 2 different approaches, Mamdani approach follows the standard Fuzzification method. So, Fuzzification method if we follow COG, it is computationally expensive, whereas the Takagi Sugeno's approach follow the simple numerical calculation and it is faster. So, we can broadly can say that Mamdani approach is easy to interpret, but less accurate and computationally bit expensive. On the other hand, Takagi Sugeno's approach is difficult to interpret, accurate; more accurate than the Mamdani approach and then calculation is fast.

So, if you want to design a fast and accurate fuzzy logic controller, then we can follow Takagi Sugeno's approach, but there is one issue, Takagi Sugeno's approach as it needs some mathematical treatment. So, whatever the rules are there they needs to be stored in the form of a some mathematical representation and that is a big challenge for the designer. So, if the designer is not so much experienced, then they can follow certain difficulty in this direction, whereas Mamdani approach is very easy to frame the rules and then rule base and then inference engine.

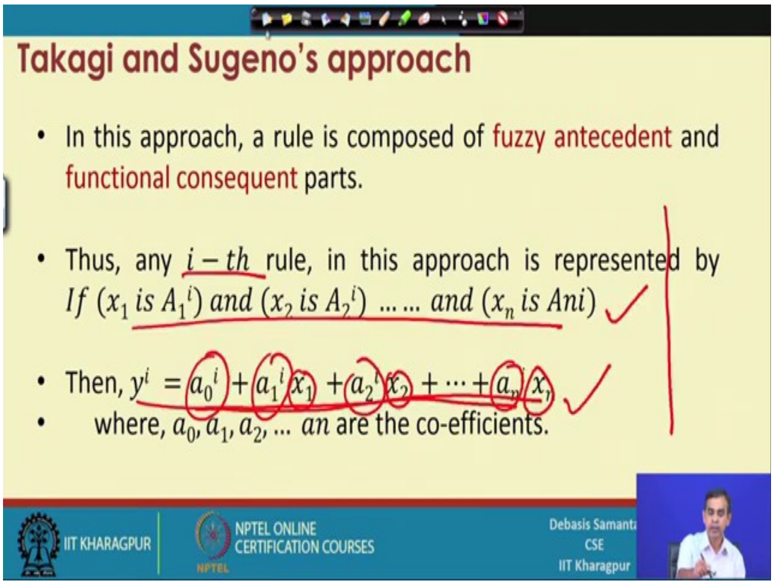
So, these are the two difference are there. Obviously, there is a trade off. Now, let us see how the Takagi Sugeno's approach so far the fuzzy logic controller is concerned and then how it works.

(Refer Slide Time: 03:15)



Now, we will consider one example, so that you can understand Takagi Sugeno's approach and mainly in terms of input example case studies rather we will explain the step, whatever the methods and technologies are there, I will discuss in time.

(Refer Slide Time: 03:29)



Anyway, let us proceed here the Takagi Sugeno's approach. Now, according to this approach, every rule is represented as it is in the same as Mamdani approach using the, if then clause.

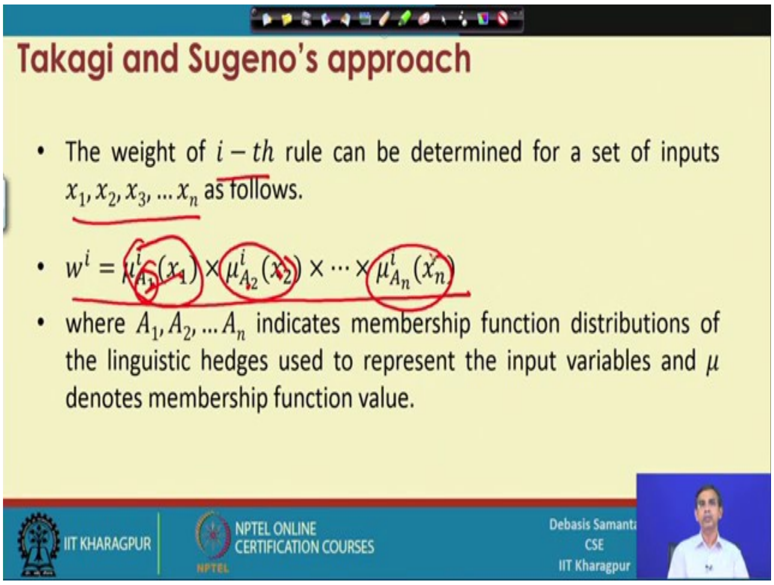
Now, for a system if there are n input, then a rule in the Takagi Sugeno's approach takes this form. It is the rule that is with the n input x_1, x_2 and x_n. So, if n input at any instant having it is own value and if these are the input, which basically related to the fuzzy state A_1, these are input fuzzy state A_2 and these are n^{th} input, the fuzzy state A_n, then the rule will take this form.

Now, this is the rule and if this is the rule, then the output of the rule also can be expressed mathematically and then for this input the output is shown here. So, it is basically input x_1, x_2, x_n and a_0, a_1, a_2 and $\ldots a_n$ are the coefficients. Now, how these coefficients can be obtained, I will discuss it. Anyway, this is the coefficient will be supplied by the fuzzy engineer based on the different rule.

So, for the i-th rule, so these are the coefficient will be totally different and this is a difficult job time to time for engineer to decide right value of this coefficient, because the output of a rule depends on the right choice of the this coefficient value. Anyway, so if fuzzy, we can depends on the wisdom of the fuzzy engineer and let the fuzzy engineer suggest these are the coefficients for the i-th rule and therefore, for any input which satisfies this, the rule we can calculate the output. So, these are the basic thing and this is very difficult job for the engineer to decide this one, once you decide it then rest of the things is very straightforward and simple.

Now, let us see how the rest of the part can be for the 2 rules, for the rules that we have discussed which takes this form like.

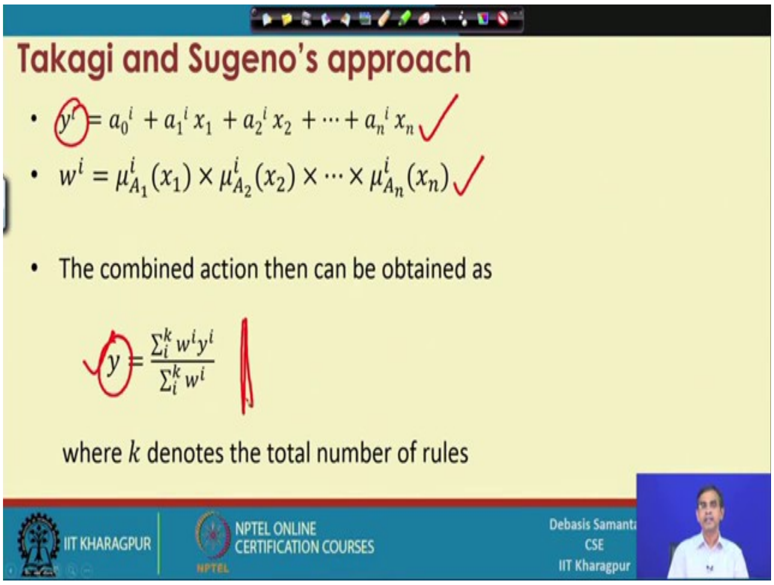
(Refer Slide Time: 05:56)



Now, according this approach, it first calculate weight of a rule. For example, for the *i*-th rule that we have discussed with the nth input, the weight; the *i*-th weight can be calculated using this formula. This formula it is like this, x_1 , which is belongs to the fuzzy state A_1 taking it is membership value, so $\mu^i_{A_1}$ and similarly for the, another x_2 which is in the fuzzy state A_2 , taking this membership value as $\mu^i_{A_2}$ and so on.

For the *n*-th input taking the fuzzy state A_n and taking it is membership value. So, it basically takes the product of the different membership values for all inputs, belongs to the different fuzzy state. So, taking these are the values, we can calculate the weight of the, *i*-th rule. So, for whatever the rules are relevant for each rule we will be able to calculate weight.

(Refer Slide Time: 07:03)



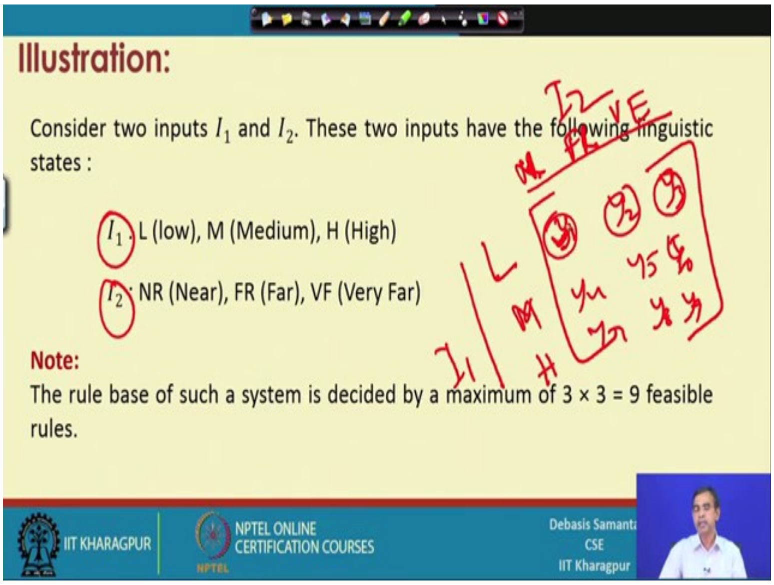
Now, after the weight is calculated, the next step is to calculate the output value, it is called the y^i. As I told you, so y^i is associated with each rule, the y^i is associated with each rule and this is y^i value for the *i*-th rule and this is a weight for the *i*-th rule, that we have calculated in the last step and taking these values for all the rules.

Suppose there are *k* rules and then taking the sum of product of all $\sum_i^k w^i y^i$ and divided by the sum of all $\sum_i^k w^i$, it will give you the final output and that is the output and you can say that this output *y*, is basically the crisp output and this is the one difference from the Mamdani approach to the Takagi Sugeno's approach, where we need to calculate the fuzzy output first and from the fuzzy output we have to calculate the crisp output.

But here, directly from the fuzzy input as the rule we can calculate the weights for each rule and then the output value of this one and then using this formula the final output, which is in the crisp value can be calculated. So, this way this is first, because straight away we can avoid the Fuzzification module there and then we can directly come into the rule.

Now I can, let us elaborate the Takagi Sugeno's approach with an example, so that you can understand. So, this is the method actually by which the Takagi Sugeno's approach calculates the output and this is basically the mechanism, where the fuzzy inference engine work. Now, whatever the rule base and then Fuzzification and everything that is the same as Mamdani approach, now let us consider, as an illustration, one example.

(Refer Slide Time: 08:48)

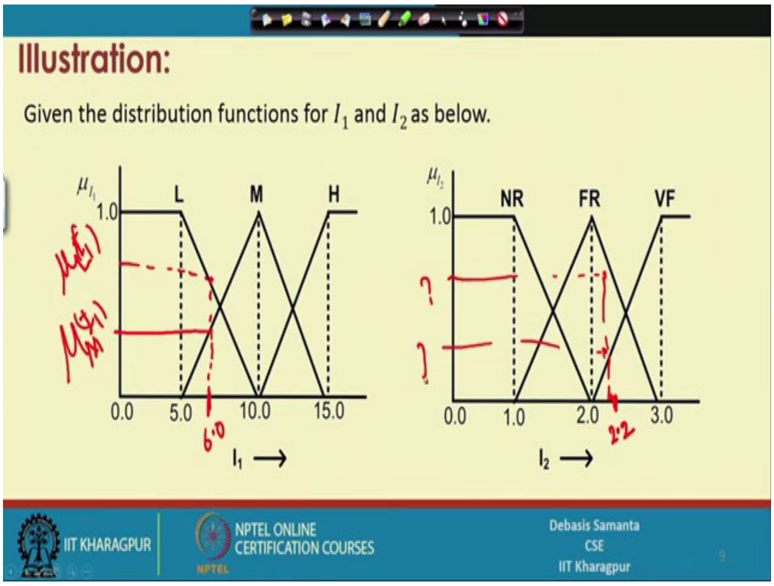


So, let us consider is a 2 input system. 2 inputs are denoted as I_1 and I_2, they let us consider there are abstract input and also consider that the 2 inputs I_1 and I_2 has the fuzzy linguistic, namely low, medium and high. That means, I_1 can be considered as a fuzzy state low, fuzzy state M or fuzzy state high with the different values of the membership for the different values of I_1.

Similarly, the other input I_2 has the 3 fuzzy linguistic again. They are called near denoted NR and far denoted FR and very far denoted as VF. So, there are 3, linguistics for the first input and then 3 linguistics state or fuzzy state for the second input. Now again if we follow the rule base matrix for this and you can understand the rule base matrix has, how many rules? Total 9 rules. So, if this is the L M and R, sorry it is a, so rule base system it is like this, so L, M and H. So, the 3 input for I_1, these are the I_1 and another 3 input for I_2. So, this is the I_2 and has NR, FR and then VF.

Now, here in this rule it is basically, so there are 9 rules. So, rule 1, the y_1 output, y_2 and y_3 similarly y_4, y_5, y_6, y_7, y_8 and y_9. Now, in the Mamdani approach, we have observed that all the rules values here in terms of fuzzy linguistic, whereas in the Takagi Sugeno's approach all the rule values in terms of mathematical representation by which the y_1, y_2, y_3 all this things can be calculated. So, if there is a rule if I_1 is L and I_2 is NR, then the output y_1, will be calculated using some mathematical notation. Now, for this again example, see what is the mathematical notation that we can consider out of these 9 rules, it is expressed here.

(Refer Slide Time: 11:29)

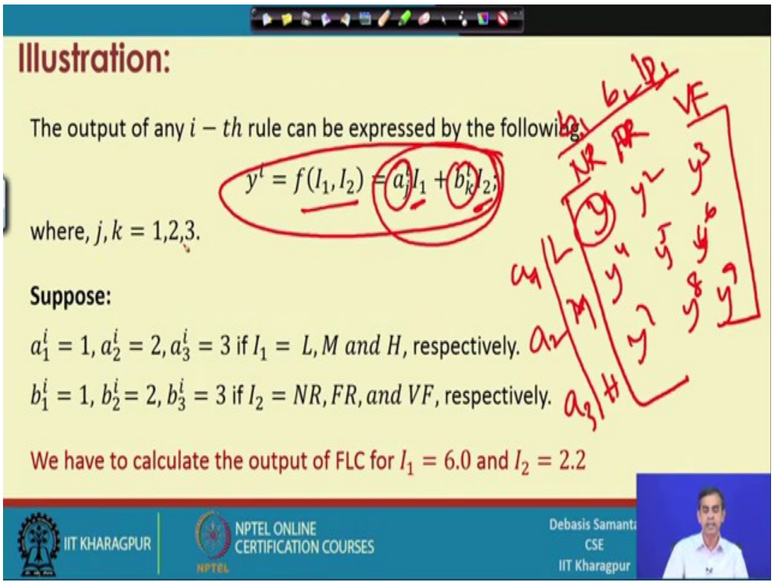


Now, before going to this, again let us consider the fuzzy linguistic state for 3 fuzzy states so far the I_1 input is concerned, which are denoted as L M and H is shown here. So, it is the fuzzy state for the L M and H, the fuzzy membership function which is like this, for L it is like this and for H it is like this. So, these are the fuzzy state defined for the, I_1 and the input of I_1, is in the range 0 to 15 as it is mentioned here.

Now, similarly for the other input I_2, the range of values that is in 0.0 to 3 .0 and above and the fuzzy state that we have considered here, it is shown here. For NR it is look like this, for FR it is like this the membership function and for VF the membership function is like this. So, this is the fuzzy state for the 2 inputs I_1 and I_2, μ_{I_1} is the membership values for the different fuzzy state belongs to the input I_1 and μ_{I_2} is a membership values for the different fuzzy state belongs to the different input as I_2.

Now, this is the Fuzzification module that is here or it is basically the fuzzy design is there. That means, all the input should be represented in the form of a fuzzy linguistic and we have discussed the 3 different fuzzy linguistic related to this illustration. Now having these the fuzzy linguistics there, now we have to calculate for a particular instant, that means, for a particular values of I_1 and for a particular values of I_2 at any moment, how the fuzzy input can crisp input can be converted to the fuzzy input.

(Refer Slide Time: 13:23)

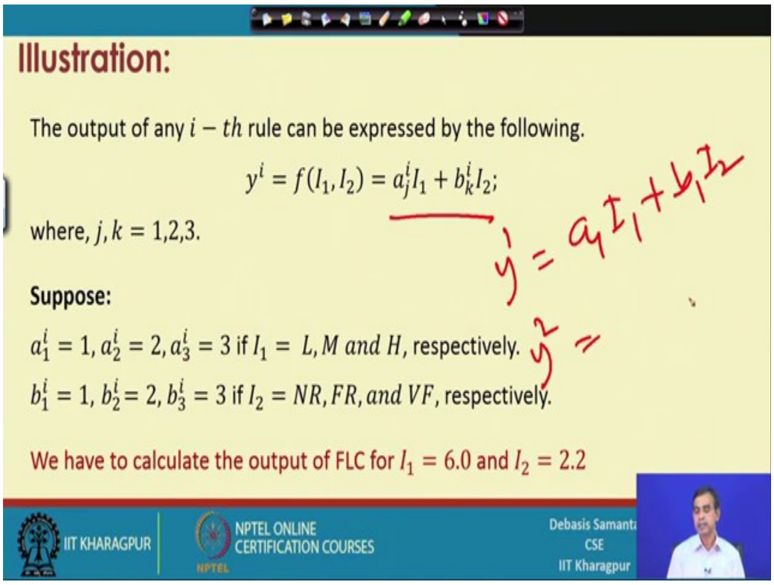


It is basically the same method that we have discussed there and we can follow it. Now here, in this slide say suppose each fuzzy rule that is denoted here in this expression. As I told you, again the same thing I can write it. So, it is basically y^1, y^2, y^3 right, y^4, y^5, y^6, y^7, y^8 and y^9, 9 rules are there. For each rule, that can be discussed in terms of the input value I_1 and I_2 and suppose we assume these are the, what is called the mathematical representation of output values.

Now, if we see, so this is the input I_1 and I_2 and then these are the coefficient. Now, all this coefficient needs to be decided by the designer. So, it is basically they are, basically how many coefficients are there L M and H. If we consider 1 coefficient a_1 and a_2, a_3 for this, similarly here NR then FR and VF, then we can consider b_1, b_2, b_3 so there.

Now, so we have to decide the different values for this rule, if a_1 is 1, b_1 is this and this one. So, there are 9 different values to be considered here, for *j* equals to 1, 2, 3 and some *k* equals to say, 9 different values to be considered there. Now for simplicity, suppose we consider the values a^i like this one, a_1 for any rule I_1 is 1, a_2 for any rule is I_1 is 2, a_3 for any rule I_1 is 3.

(Refer Slide Time: 15:26)

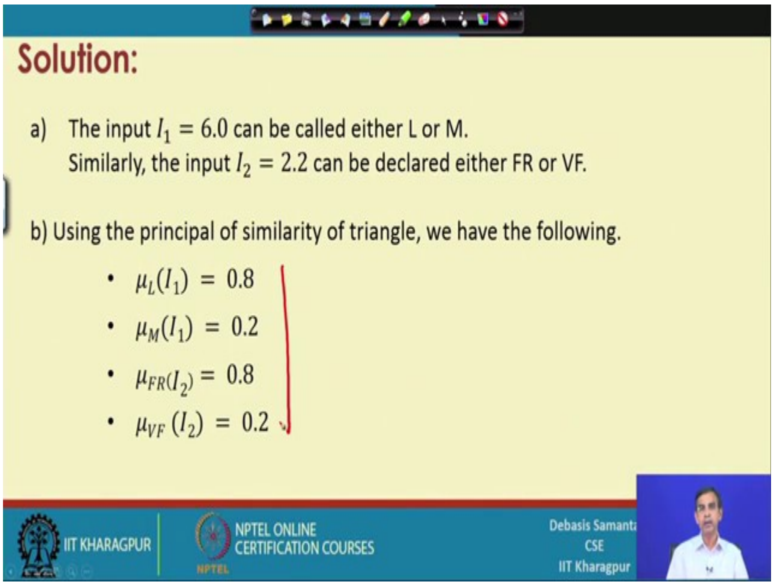


So, you have a simplified assumption and suppose it is decided like this one. So, alternatively I can say like, y^1 this is equals to $a_1 I_1 + b_1 I_2$ this is the 1 rule like. similarly y^2 there is another rule also can be considered likewise. So, all this expression, that means, for the output can be represent like this. Now, having this is the representation of a particular rule and corresponding to that particular rule basically the output value, compress the rule base for the fuzzy system according to the Takagi Sugeno's approach.

Now, let us consider a particular instant for a certain values of I_1 and I_2. We consider, this is the input at any moment, that I_1 is 6.0 and I_2 is 2.2. Now given this is the input, we have to calculate the fuzzy output according to the Takagi Sugeno's approach. Let us see, how it can be calculated? Now, again for this $I_1 = 6.0$ and $I_2 = 2.2$, we have to calculate the μ values of the member elements. That belongs to the 2 different fuzzy state. Here, so $I_1 = 6.0$, so this basically this is a 6.0 and for the I_2 we have considered 2.2, so it is the 2.2 right. So, if this is even, then so this basically 6.0 belongs to that 2 fuzzy state M and as well as another fuzzy state L. So, I_1 is 6.0 belong to the fuzzy state M, has the $\mu_M(I_1)$ value this one.

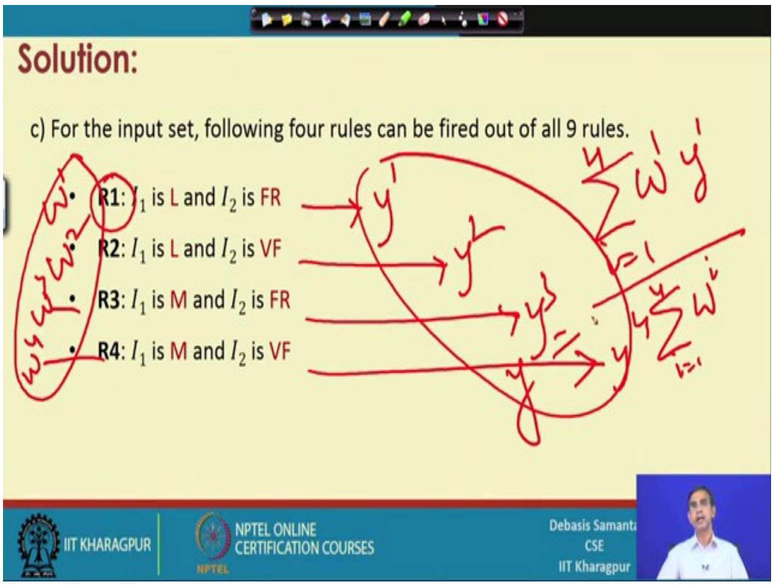
So, it is basically $\mu_M(I_1)$ value this one. Similarly, the 6.0 the input I_1 belongs to the fuzzy state L, having this value. So, it is basically $\mu_L(I_1)$ is the value. Likewise, 2.2 has the 2 fuzzy state namely, VF and then another is FR 2 1. So, this value and this value are to be calculated. So, this μ values can be calculated from this fuzzy description of the state are the same way, as it is followed there in the Mamdani approach, using the principle of similarity of triangles, we can calculate it. So, this μ values will be obtained. Once the, this μ values are known to us, then we can calculate the output value easily.

(Refer Slide Time: 18:06)



Now here, for an example we can calculate the μ value this is for the 2 input, this one as this one, $\mu_L(I_1)$, $\mu_M(I_1)$ this one and this one.

(Refer Slide Time: 18:08)



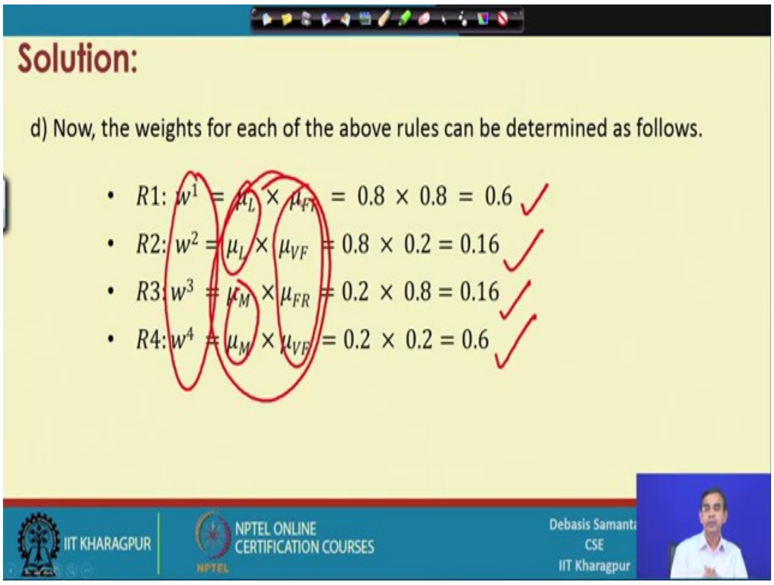
So, this are the 4 different values that can be calculated from the fuzzy linguistics state description and using the similarity of principal that we have already discussed about in Mamdani approach.

Now, so these values are known to us now. Now, we will use these values and then the output values for each can be calculated and so this is the method that can be calculated here. Now, here you can see, again just like Mamdani approach there are 20 rules, here 9 rules. Now, out of the 9 rules we have to select those rules, which are basically fireable. Fireable means, those rules are relevant in the context of current input. Now, as you see the, for the $I_1=6.0$ and $I_2=2.2$ and when $I_1=6.0$ it belongs to the fuzzy state L and M. Similarly, when $I_2=2.2$, it belongs to the fuzzy state FR and VF.

So, putting all this things together, there are therefore 4 rules, the 4 rules are discussed here. So, if I_1 is L and I_2 is FR, then it basically gives y^1 , 1 rule. Similarly ***R2*** gives you y^2 and this one gives you y^3 and this gives you y^4 . So, the 4 rules and related to the 4 rules, the 4 different output can be obtained. So, the task, the object, the next task is basically for each rule we have to calculate w^1 . These are w^1 that is the weighted value of the rule, for this rule also w^2 , for this rule w^3 and for this rule w^4 .

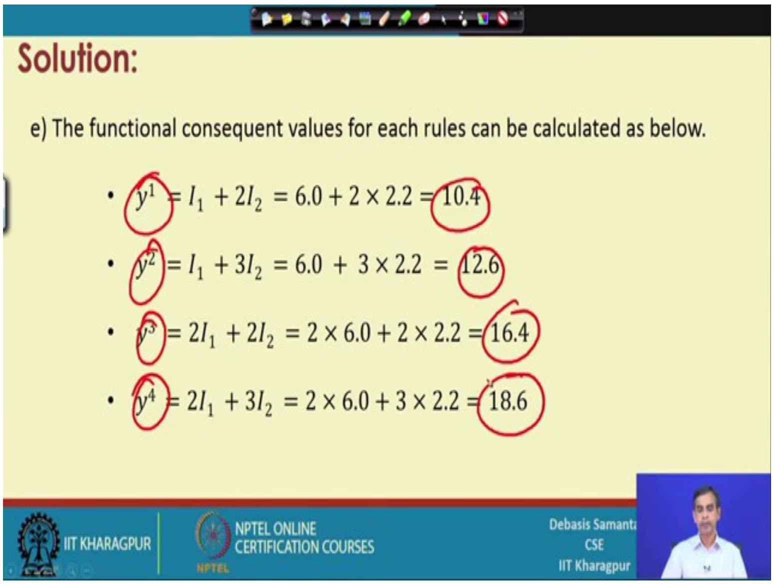
So, we have to calculate these are the values so far the output is concerned and these are the values so far the rules strength is concerned or weight is concerned. Once, we have these values, then we can calculate $\sum_i^4 w^1 y^1$ is a numerator and then the summation of $\sum_i^4 w^i$. So, this will give you the final output according the Takagi Sugeno's approach. Now, let us see in the current context, how the results can be obtained for the 2 things are there. We have considered the different values of a_1 and b_1 in the context of this example.

(Refer Slide Time: 21:00)



Now here, so this basically shows the computation of the weights value, as I told you it is a simple product of the membership value is there. So, for the rule 1 the value that can be obtained is 0.6, for the rule the value that can be obtained 0.16 and like this one. So, these values can be obtained based on the different values of the membership values in the context of current input. Current input here or here so far for the I_1 is concerned and these are the I_2 is concerned. So, w^1 can be calculated. Now, once the w^1 is known, then we will see how the y^1 for each rule, ***R1*** and ***R2*** can be calculated.

(Refer Slide Time: 21:54)

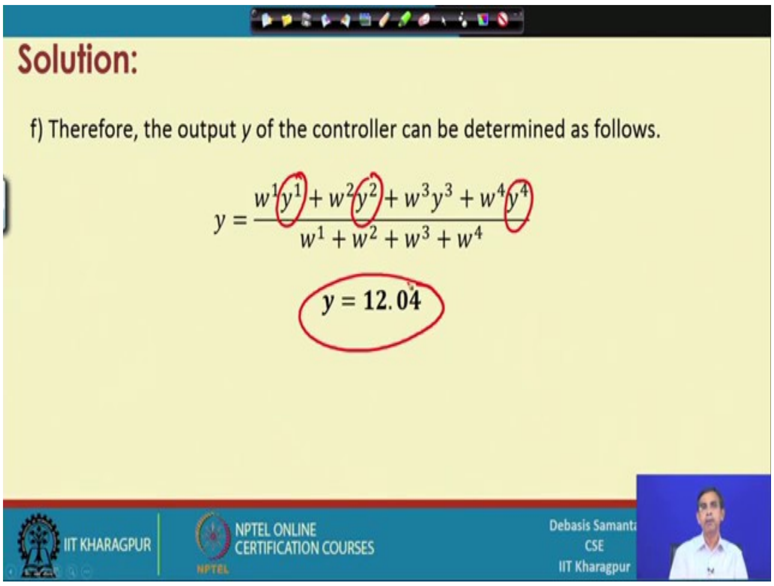


So, here is the method by which the y^1 that means output value can be calculated. I told you that, we will consider the different coefficients. For the rule 1, we take a_1 is 1 and b_1 is 2 and for others, a_2 is 3; a_2 is 2 and b_2 is 3. So, these are the values of rule can be calculated like so for the different values. So, here basically a_1 is 1 and b_1 is 2 and in this case b_2 is 3 and a_1 is 1.

So, these are the, basically the rule mathematical representation of the different rules are there and then we can obtain the values of the outputs using this. This is basically the mathematical representation of the rule 1 and this is a mathematical representation of the rule 2 and mathematical rule 3 and rule 4. So, I have given this as an example, but actually the, these are the tricks by which this rule can be decided or defined in the system and once this rule is defined according the system, it is a straight forward to calculate in terms of the different value. For example here, I_1 is 6.0 and 2.2 and taking the *b* values 2 1. So, this rule I_1 can be calculated this 10.4.

So, this is the, I_1 ; y^1 the value is this one and for the, I_2 the value will be there and I_3 the value will be there and y^4 the value will be there. So, 4 rules related to I_1 is 6.0 and I_2 is 2.2 can be calculated, their corresponding output values. We have calculated the weights of all these rules also. So, these values then can be used finally to calculate the final output.

(Refer Slide Time: 23:51)



So, the final output is the formula according the Takagi Sugeno's approach. That says that, it is basically product of the weight, weighted output values. So, it is basically, these are the sum of the product of weighted output values, divided by the sum of the weights and if we follow the calculation as we have obtained in the last calculations, then finally the y value that can be obtained as this one.

So, that straight forward way from the rule base we can select the relevant rules and from the relevant rules we can calculate the weights of each rules and then output values of each rules and using this expression according the Takagi Sugeno's approach we will be able to calculate the output value and once the output value is known then will we can take a decision and you can note that this is the output that we have obtained is crisps. As I told you, no need to do any, follow any defuzzification method in this case state way.

Here, basically state way actually, this is because state way for each rule we have taken the crisp value. All these y^1, y^2 are all these are the crisp value for each rule. That is why, state way we can obtain the crisp value as the final output. So, this is the method that is followed there in Takagi Sugeno's approach and so this is the way that the 2 controllers, the Mamdani approach and the Takagi Sugeno's approach can be designed and 2 approaches have their own advantage as well as the disadvantage and as I told you, so Mamdani approach is easy to interpret, easy to interpret means rules are interpretable. On the other hand, the Takagi Sugeno's approach the rules are difficult to interpret, so less interpretability. Difficult to interpret because, it is expressed in terms of some mathematical formula, which needs some coefficients to be design decided.

Now, deciding the coefficient it basically is a task for the fuzzy engineer or fuzzy designer. If the fuzzy engineer can decide the coefficient correctly or accurately, then it will give the accurate result. So, accuracy of the fuzzy logic controller according to the Takagi Sugeno's approach solely depends on the performance or experience or prudence of the fuzzy designer. That is the only thing; otherwise, fuzzy logic controller according to the Takagi Sugeno's approach is fast and more accurate, compared to the Mamdani approach.

So, it is up to the fuzzy designer, which approach he wants to follow, if it is a critical application, where accuracy and then the speed of the controller is very important, then they should follow Takagi Sugeno's approach. Otherwise, both the approach are equally applicable to design any fuzzy logic controller. So, this is the 2 controller we have discussed and then using this the concept, we can follow any fuzzy system can be designed from the fuzzy logic. So, this is end of the fuzzy logic discussion and we will study about some case studies that will be given as a some special problem for your practice.

Thank you.

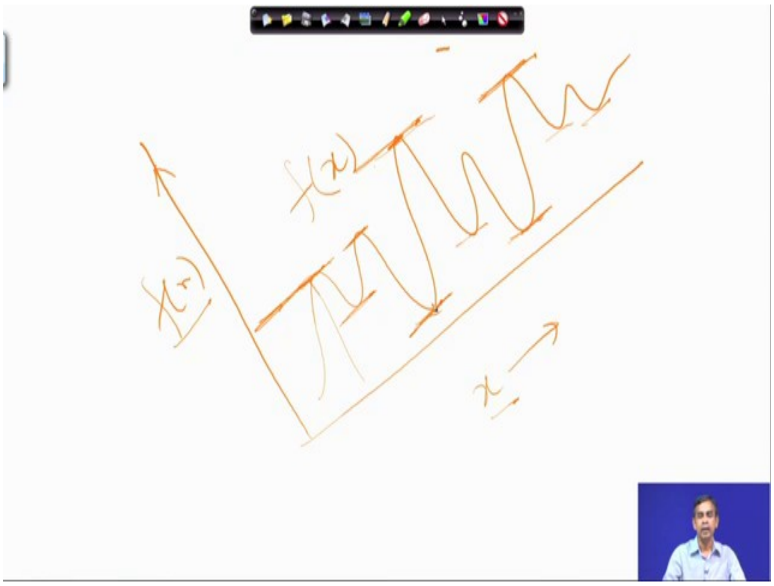
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 14
Concept of Genetic Algorithm

So, there are three computing paradigms, which is followed in soft computing one is fuzzy logic, another is genetic algorithm and the third is artificial neural network. So, today, we will introduce the genetic algorithm the computing paradigm for soft computing. Now genetic algorithm is basically used for solving optimization problem. Now you know exactly, what is an optimization problem.

So, the optimization problem essentially solving the to find the optimum value. That means, find the minimum or maximum value. As an example say suppose, this is the value for which the different.

(Refer Slide Time: 01:05)



So, suppose $f(x), f(x)$ is a function which varies with x. So, if it is x in this direction then $f(x)$ this is the $f(x)$. Now, so this concept say suppose the value of $f(x)$ varies with the x, using this form, it is like this now. So, this means the value of $f(x)$ varies with x and it takes the form like this and you can say that x as it varies then it has many values.

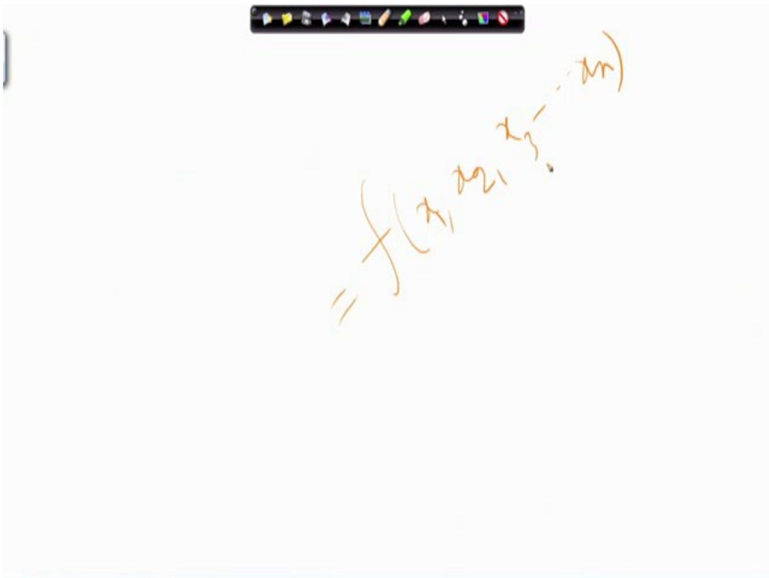
So, here basically the for some x, the value $f(x)$ is highest, it is the maximum, it is the another maximum or so on. Similarly, minimum these are minimum this one. So, it basically says that how the $f(x)$ varies with x and if it is like this and if you have to find which is the maximum value for which or what is the maximum value of $f(x)$ or for which values of x, it is the maximum.

So, that can be obtained and if you have to search it then it is called the searching for an optimum result and. So, far the searching for an optimum result is concerned, it has many what is called the values actually, here these are the value is a highest value is a peak. All peaks are basically; some values they are called maxima. Similarly, all this is the lowest values the valley it is called the minima.

Now, out of this so many maxima's there are some values it is called the global maxima, if we say that this is the maximum of all the maxima values. Similarly, it is a local maxima local minima if it is minimum of all the minima values there. So, the concept is local maxima or local minima and then global maxima or global minima. So, finding an optimum value; that means, a minimum which is global maxima or global minima is called the solving for optimization problem.

Now, we will discuss about the GA. The genetic algorithm, which basically gives us a unique and fantastic way to search for all optimum values. That means, either minima or maxima before going to discuss about the genetic algorithm, we will just discuss about, how mathematically the optimization problem is defined.

(Refer Slide Time: 04:09)



If suppose $x_1, x_2, x_3 \ldots x_n$, are the input parameter is given to us and we have to find, the function which is defined by f and which is discussed with this in terms of these are the input parameters. That means, the value of f is decided by the value of all these parameters $x_1, x_2, \ldots x_n$. So, this is a function we call this function as an objective function. So, this function is called objective function.

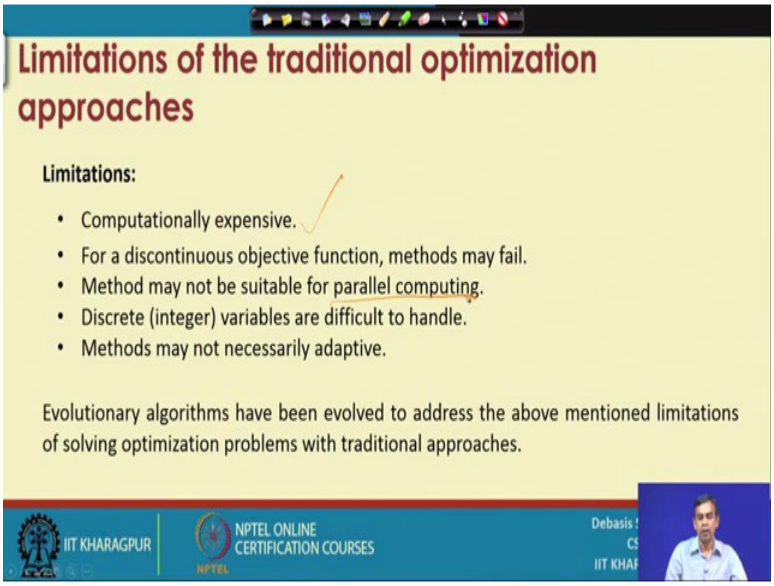
Now, we have to find either say optimum value, optimum means either minimum value or maximum value, for a given set of values of these $x_1, x_2, \ldots x_n$. Then this is called the optimise value. So, this optimise value can be either minimize, if we have to find the minimum values or we have to either find maximize if we have to value the maximum maximization value.

So, objective function always in terms of either minimization or maximization and this basically define by means of an objective function, which takes like this $x_n . N$ ow this functions this optimization. In fact, subject to certain constants. So, if it is minimize and constant may be another function g_i ¿ $x_1, x_2, \ldots$ same set of parameters.

Now, this may be equals to 0, where they are may be one or more constant. So, this i, i = 1¿ m, so if m constants are there. So, these basically this is the objective function it is like this and this is basically the constant usually we denote as subject 2. So, there may be $g_1, g_2, \ldots g_m$ constant. So, here the finding an optimum value; that means, finding some values of the input parameter. So, that this function returns the optimum value and it should satisfy all these constant.

Now, this problem is no more trivial problem. In fact, and this problem cannot be solved in normal time, that is why we need some pragmatic approach like say soft computing to solve the optimization problem. That means, we have to find the values of input parameters for which a function f should return on optimum value minimum value or maximum value and at the same time, it should satisfy the search constants number of constants. So, this program is no more a simple program.

(Refer Slide Time: 07:28)



So, traditionally there are many methods are available and run to solve this kind of optimization problem, but they have their own limitation actually the traditional optimization methods are in fact, computationally expensive. That mean, they cannot be applied to solve some optimization problem in real time. It may take 1 month to solve some problem like and all the traditional optimization methods usually not suitable for a discrete objective function and.

So, they are not suitable for discontinuous objective function there are some functions which have some value in some ranges. So, is a discontinuity they are in the objective function those method fails and as it is the time consuming, what is called the task finding an optimum value. So, usually we advise to follow parallel computing, but the parallel computing may not be implementable may not be realise using traditional optimization.

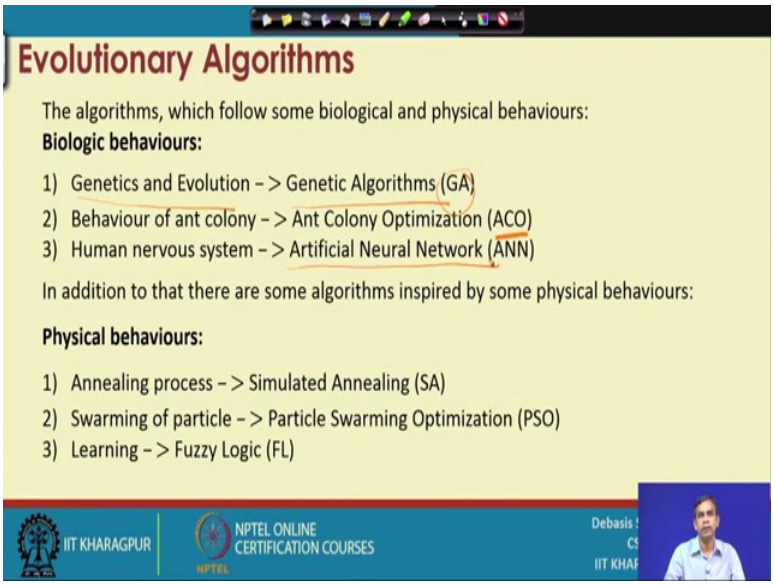
So, we need something which basically suitable for parallel computing and it is observed that traditional optimization approaches are not good enough to deal with the discrete values of the input parameters. So, if the input parameters are having discrete values then the existing optimization technique cannot solve them, and another limitation of the existing approaches is that they are not necessarily adaptive.

Adaptive in the sense that the same algorithm that you have developed, if you have to apply to m number of parameters instead of n , where n maybe ¿m or ¿m ; that means, if the input parameter increases then you have to rewrite the method the program totally differently. If the say, input parameter type is different an earlier it was for integer.

Now, you have to see it real type so, then it cannot be. So, they are actually they are not adaptive; adaptive means if the environment changes, input changes, that input parameter type changes then the traditional approach is not easy to adapt them.

So, we need some new method, which basically address all these limitations and we will see the evolutionary algorithm, it is an alternative approach to the traditional optimization approaches that can solve and then address all these problem. So, genetic algorithm is basically one special type of evolutionary algorithms.

(Refer Slide Time: 10:13)



Now, so far the evolutionary algorithms are concerned how they are different than the traditional approach. So, they are different in terms of the way they solve the problem. In fact, the evolutionary algorithms they follow few concepts and the concept is called the, they follow certain biological and physical behaviours, which is around our globe in our world.

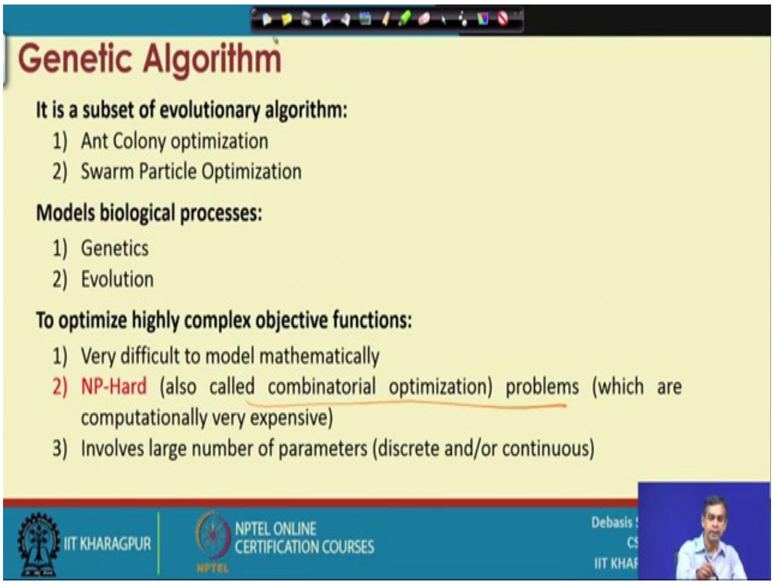
So, genetic algorithm which we are going to discuss is basically follow, the concept of genetics and evolution. Genetics is a well-known concept in biology and evolution is also an well-known concept in biology. So, these are biological concepts. So genetics and evolution is followed to solve the optimization problem then this is called the genetic algorithm and popularly it is abbreviated as GA.

Now, the way the ant they collect the food or they invite others fellows to a particular food source. It has been, followed to solve optimization problem and this is called ant colony optimization. So, it is also some sort of behaviour of ant, which has been followed and their behaviour is basically adapted into solve optimization problem it is called the ant colony optimization called ACO.

Now like there are how our nervous system work, if we follow the concept then and if we apply it then you can solve any problem this is called the artificial neural network or ANN. So, these are the classes belong to the biological behaviours there are some physical behaviours, the matters how they work. Now annealing process is the one process which is used to prepare the metals and if we follow the annealing process to solve a type of optimization problem or optimization problem, then it is called the simulated annealing, it is abbreviated SA.

Now particle how this swarm in a stream or flow the same concept can be followed to solve another optimization type of problem is called the particle swarming optimization problem or PSO. We have learned about fuzzy logic; how fuzzy logic can be used to learn. So, this is also another physical behaviour. So all these concept are basically the concept, which is followed in evolutionary algorithms.

(Refer Slide Time: 12:57)

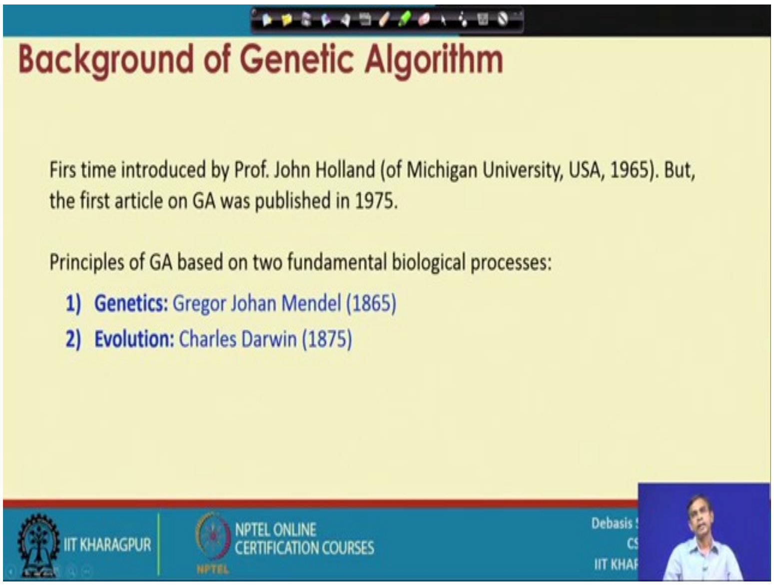


Now, in this lecture we will basically focus on specific evolutionary algorithm, it is called the genetic algorithm, As I told you genetic algorithm like ant colony optimization, particle swarm optimization is another type of evolution algorithm and it follows the two important biological processes called the genetics and evolution and particularly it has been observed that genetic algorithm is tremendous successful, in case of solving the problem which are basically called combinatorial optimization problem; that means, the problems which cannot be solve in real time.

It is also called NP-Hard problem, or you can say that additional methods if we apply to solve this kind of problem. That means, NP-Hard problems it is computationally very expensive and cannot be computed in real time. So, it is the problem and then genetic algorithm have been applied to solve this kind of problem and we can see the result in real time. And more significantly, the genetic algorithm is best suitable for those kind of problem for which any specific mathematical model or a suitable algorithm is possible to define how to solve the problem. If we are not having that specific algorithm are steps to solve the problem, when we can apply the genetic algorithm to solve this kind of problem.

So, the problem which is very difficult to module mathematically or specific algorithm is available then we can apply the genetic algorithm to solve this kind of problem. And if a problem involves a large number of parameters, the parameters maybe discrete or maybe continuous or anything then the traditional approach is very difficult to use it, but genetic algorithm can be used to solve this kind of problem efficiently and effectively.

(Refer Slide Time: 15:04)

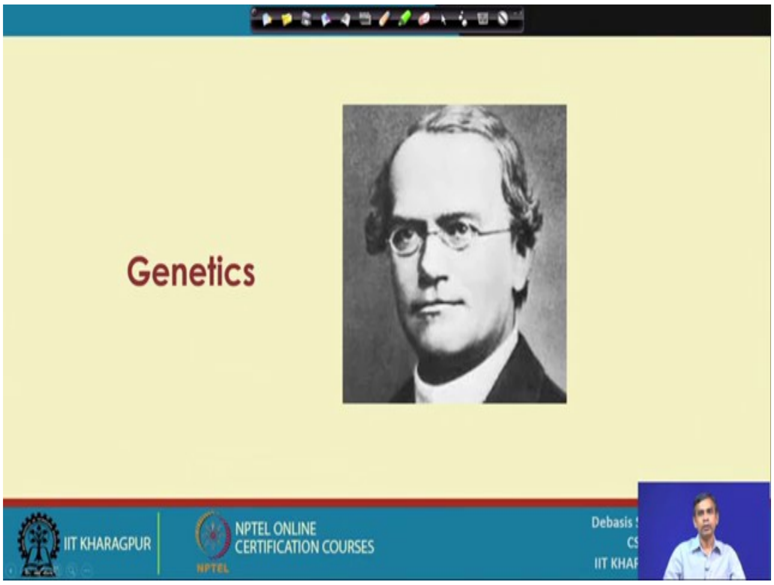


So, this is the idea, this is the history behind the genetic algorithm and which we can follow it. Now I just want to start with a little background about genetic algorithm. So, it is as early as in 1965, Professor John Holland, from Michigan university. He first proposed the concept, concept of genetic algorithm in 1965, although he has proposed the idea, but ultimately it was acceptable to the research community much later in around 1975.

In fact, the two pioneer who work to make the GA, most successful they are the two revolutionary people one is called the Gregor Johan Mendel and Charles Darwin. Gregor Johan Mendel in 1865, he proposed one revolutionary concept called the genetics and around 10 years later, Charles Darwin who proposed the concept it is called the evolution and that these two concept are merged together to solve the optimization problem which become the true I mean origin of the genetic algorithms.

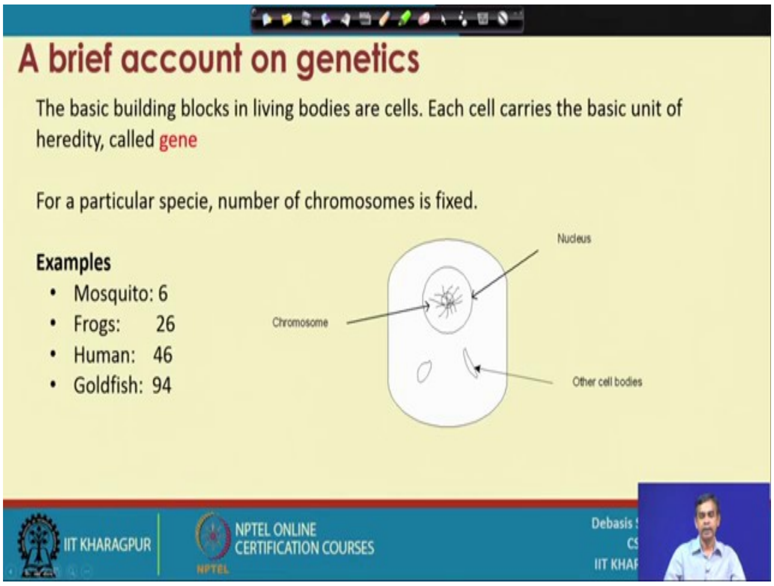
So, in order to learn the genetic algorithm, it is better that we should learn about the two things the genetics and evolution first.

(Refer Slide Time: 16:29)



Now, as I told you Gregor Johan Mendel is the forefather of the concept of genetics.

(Refer Slide Time: 16:34)



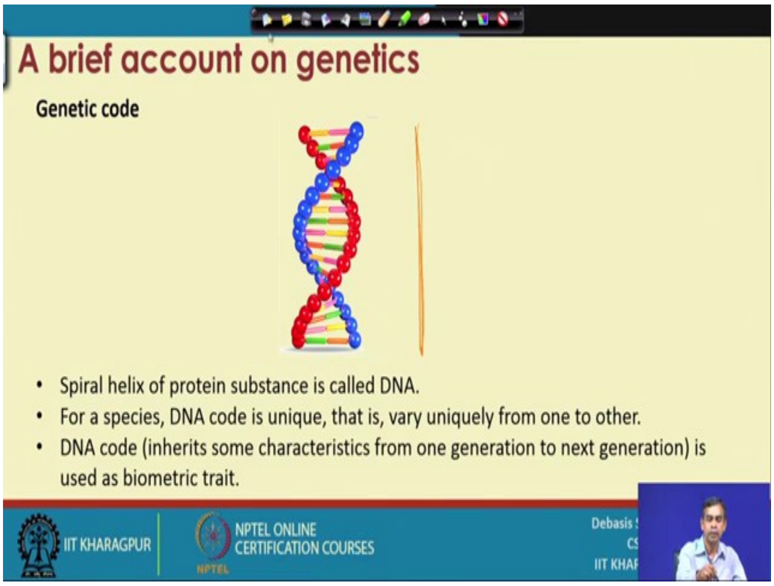
And genetics is a well-known thing, and you know genetics came from the concept called gene and gene is basically being a fundamental thing in our life, and it basically say that our body is consist of a large number of cells, living cells and each cell is basically a consists of what is called the one essential part in the cell it is called the chromosome, and if we go into details about the chromosome.

In fact, there is a spiral helix form and they are called the genes and these genes are basically the characteristics of a particular cell. So, or in other words a chromosome decides a particular type whether it belongs to monkey or it belongs to man or it belongs to cow.

So, it is also observed that chromosome in terms of number the different species that we are having had the unique number of chromosome. For example, mosquito has number of chromosome 6, human has 46, 23 pairs and goldfish 94 out of is goldfish is the one element which is having the largest chromosome.

Now, so chromosome is one important be things that is there and. In fact, this chromosome also plays an important role in our genetic algorithm, we will learn it exactly how the chromosome is synonymous to genetic algorithms. First let us see exactly how the chromosome actually works it.

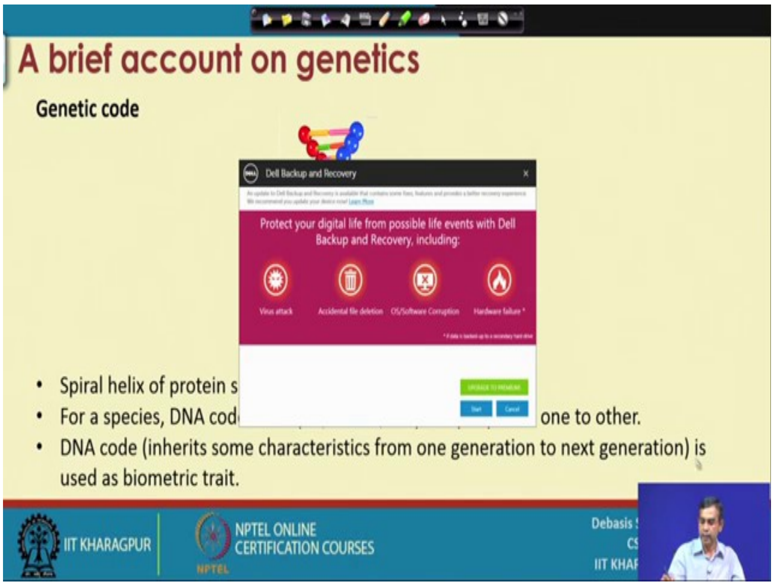
(Refer Slide Time: 18:04)



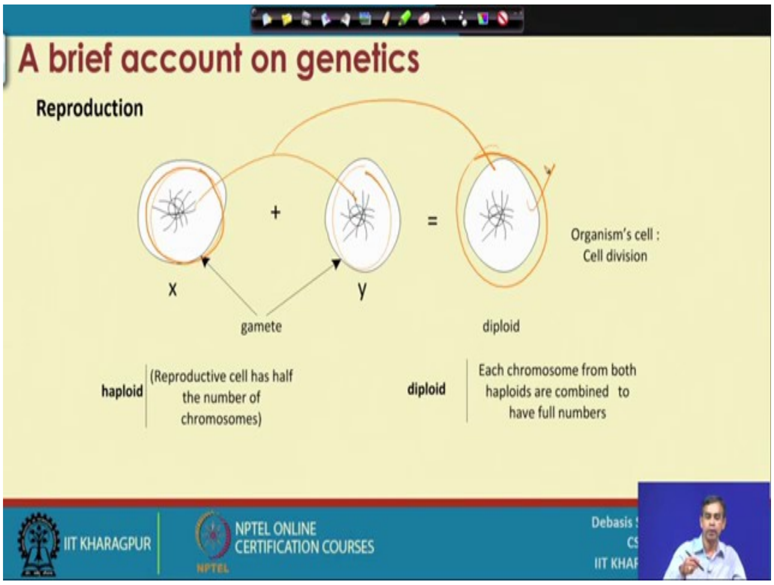
Now see, chromosome basically is a code it is called the genetic code also and we know that every individual has its own characteristics, own features, own specification this is because the genetic code is unique and it is different, it is complete different from any other individuals around.

So, a genetic code is basically looks like a spiral helix, it is basically a protein substance and this protein is called DNA, deoxyribonucleic acid and a typical look of the protein DNA is look like this. So this is a DNA structure and this DNA, has its own unique structure for a particular individual and that is why you say that it is unique, and if we can represent this DNA code then we can basically identify the person. In fact, so that is why this DNA code is used as a biometric trait that mean by this DNA, we can identify a person uniquely.

(Refer Slide Time: 19:13)



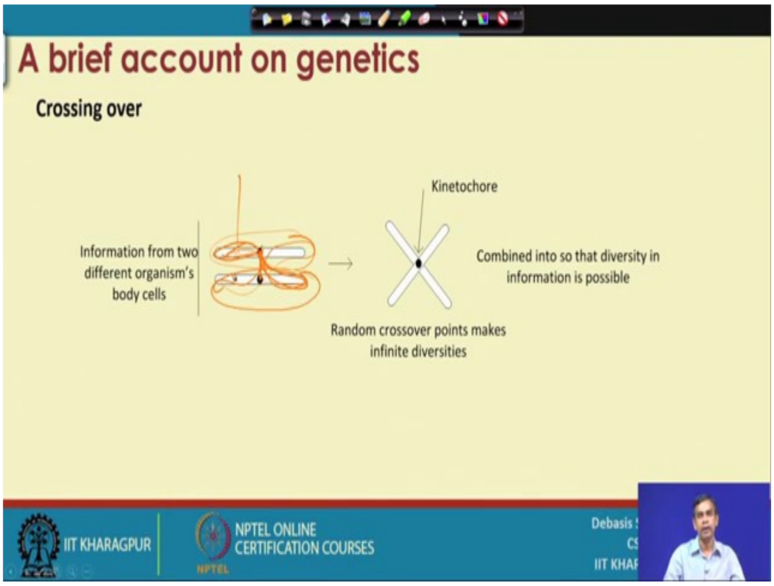
(Refer Slide Time: 19:16)



Now, so this is a concept of DNA and then the, this concept is basically, is also important in the reproduction, the reproduction as you know. So, that two what is called a half cells they are called haploid, to half cells from the two opposite sex male and female obtained and then when they merged they form the diploid and then there this diploid is basically form a cell.

So, here very important thing is that haploid is basically one part which has the half number of chromosomes and this another half number of chromosomes, and when they merge together they form the diploid which basically form the full number of chromosomes. So, here is basically the division after that unification and then it produced another unique. What is called the unique identity or unique elements.

(Refer Slide Time: 20:14)



So, this is the concept that is followed in reproduction and so this is a part of the life and we just follow the reproduction. But behind this reproduction there is one important thing that we have learnt about that from two haploid we got a diploid and here is the idea about. So, this is the one chromosome from one haploid another chromosome from another haploid and there is one point, it is called the kinetochore point.

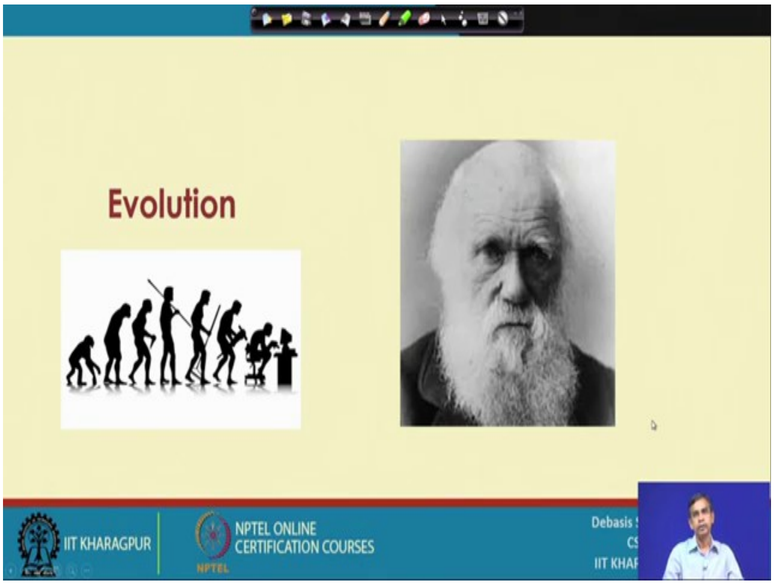
Now, they basically combine this kinetochore point and then. So, from one element here and another element here and then if we consider another element which is basically one part of here and another part here. So, basically it, basically gives the one chromosome to diploid. Similarly, another chromosome to diploid. Now, here one thing you can note if we take one part here and one part here.

So, the new chromosome that we get, so it has the two call a mixture of chromosomes and that mixture of chromosomes basically if produce is able to produce a new elements or new identity.

Now, so this is the fundamental thing that is followed there, if we follow these kinetochore points in different position. Then we can have the infinite number of different possibilities of having the different unique identity. So, in this sense the reproduction allows or reproduction produces a unique element, every time it reproduces from two chromosomes to another chromosome or two haploids to another diploid.

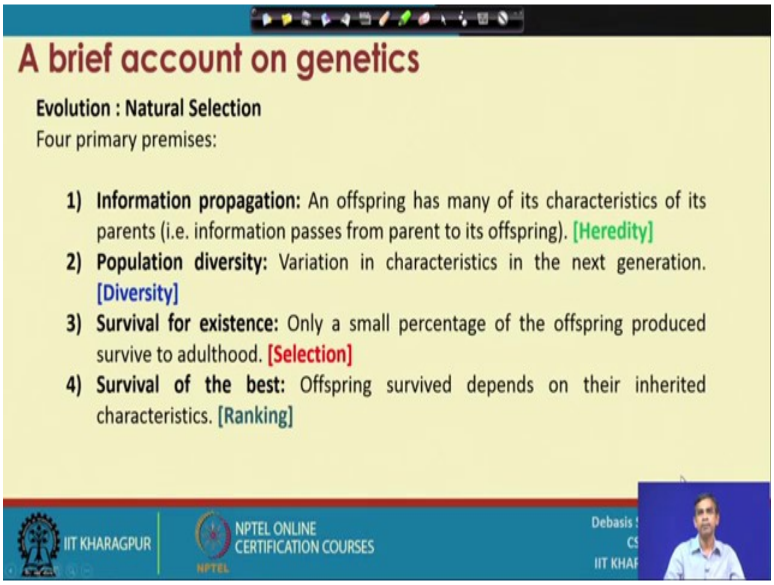
So, this is idea that is follows there and this concept in genetics is called the crossing over and we will follow it exactly the concept of chromosome as it is there in genetics. Similarly, the crossing over or simply it is called the crossover is also an important what is called the philosophy that is followed in optimization technique.

(Refer Slide Time: 22:09)



Now, so this is the genetical genetics, which basically Gregor Johan Mendel proposed first then how reproduces and every reproduction produce the unique element. Next evolution is basically improvement from one level to another. So, regarding this in evolution the Charles Darwin is the forefather of this what he proposes four concepts. So, far the evolution is concerned the four concepts.

(Refer Slide Time: 22:38)



Are basically heredity, diversity, selection and ranking, so according to the Charles Darwin heredity? It is basically called information propagation, information propagation means that an offspring has many of its characteristics of its parents and therefore, the property or characteristics from its parent is basically passes through its offspring; offspring means children.

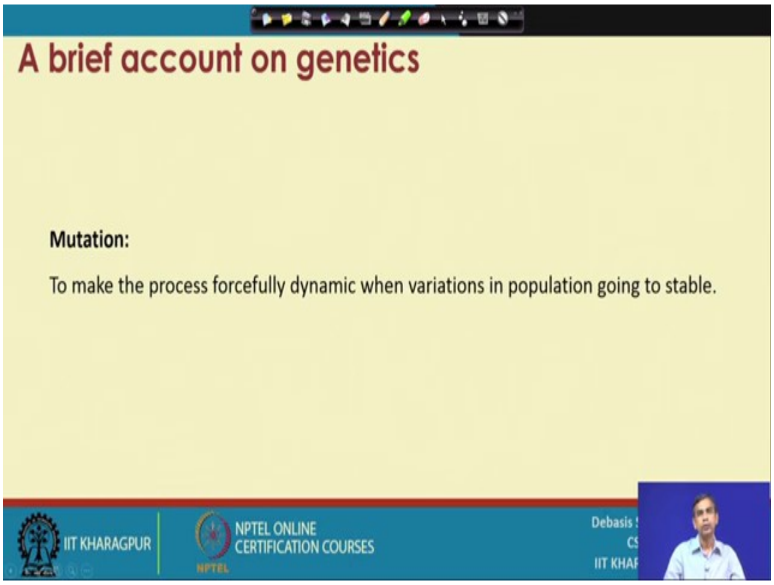
So, this is called the heredity. That means we inherit something from our parents. So, that is the concept heredity and population diversity. So, Charles Darwin termed it as diversity only, so it is basically called the variation in characteristics in the next generation.

So, if we see the different generation no two generation we can obtain which have the same identity always it has at least some minor difference maybe, but differences are there. Next premise is called the selection, that is very important and this selection Charles Darwin termed is at survival for existence. So, basically out of many offspring only a small percentage of the offspring is basically able to survive in to adulthood and other basically go to I mean dies to exist, there they cannot sustain much more.

So, that is the selection and our world is basically followed, this selection procedure and that selection is basically called the survival of the best. So, Darwin call is a survival of the best. That means, only those offspring they survive depends on their inherited characteristics. So, it is based on the ranking, so these are the four things, four premises rather which is followed they are in so far the evolution is concerned.

So, evolution will be carried forward and Charles Darwin shows that these are the four primary things by which the evolution can takes place and evolution is followed there and this evolution also Charleston called the natural selection initially so, but will termed is an evolution.

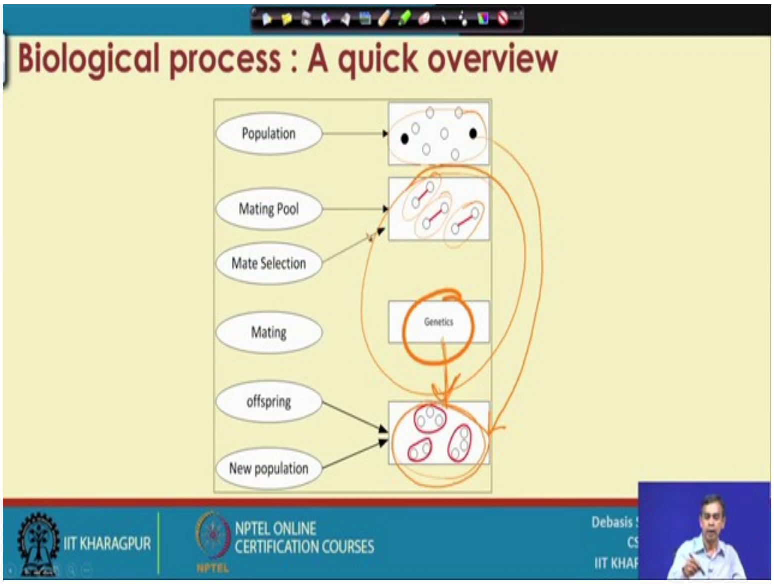
(Refer Slide Time: 24:46)



And we will see exactly how these two concepts are followed in genetic algorithm and then genetic algorithm has been proposed. Now, other than this genetic algorithm concept, there is another is called the mutation will discuss about the mutation a mutation means all of a sudden there are some changes. So, two parents those are the fair skin all of sudden their offspring may be black also.

So, it is the due to the mutation all of a sudden there are some changes. That means, there are certain drastic differences in their chromosome property and that is the mutation. So, mutation is also one part of our natural what is called the process and natural generation production or genetics.

(Refer Slide Time: 25:30)

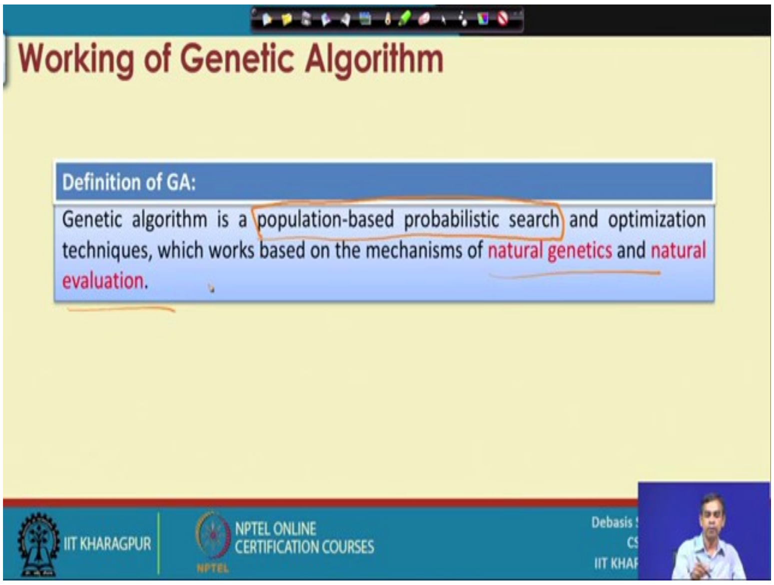


Now, so this concept is basically followed there and will learn about that how the concept of biological process namely genetics and evolution is followed there I just briefly summarise the concept that we have learned so far. So, if we have a population which is the population initially and then from this population, we follow the mating pool, mating pool means we just simply see that, who can be that can be fitted for mating another's.

So, there is a mating selection like marriage or whatever it is there. So, after the mating selection is there then it starts a mating and is basically. So, genetics is followed there. So, that is a mating and then from the mating, we have we follow whatever the crossover mechanism or cross crossing over that then and they produce there they produce the reproduction.

So, this reproduction produces, the new offspring and new offspring all together produce the new generation. So eventually the idea is that from the current population following the reproduction procedure, we obtain the new population and here in between there are genetics and evolution involved. Now these are the concept that is followed there and in genetic algorithm. So, we are basically using the same concept.

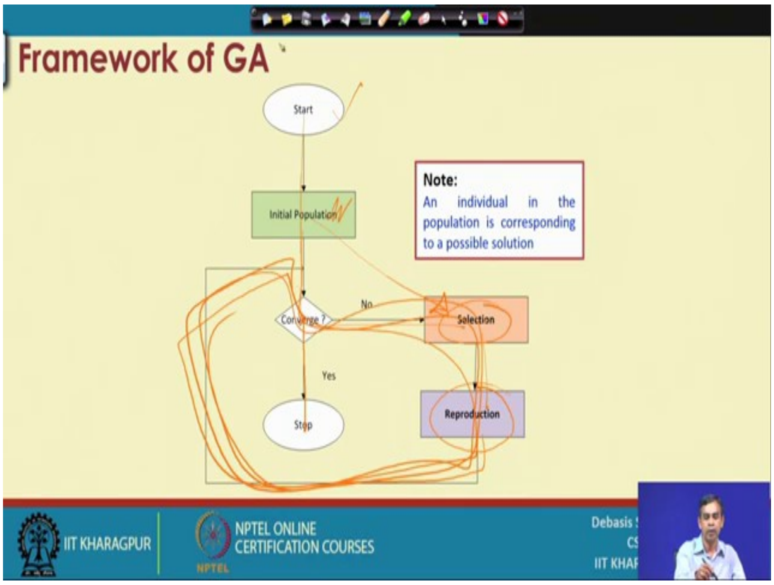
(Refer Slide Time: 26:47)



So, I can start with the genetic algorithm concept, it is basically the is an algorithm and this algorithm is a population based and is a probabilistic search, probabilistic search means the mating and the reproduction is a probabilistic random and then optimization. It is basically selecting the best candidate, from there and which works based on the concept of genetics and then evolution.

So, this is a concept of genetic algorithm so the fundamental thing it is there the genetic algorithm is basically, is a population based probabilistic search that is important. Now we will learn about how the population based probabilistic search can be achieved and using the genetics and then the concept of evolution. So, this is our objective study objective and we will see exactly how this can be done.

(Refer Slide Time: 27:42)



Now, quickly I will start with first the architecture genetic algorithm. So, this is basically the flow chart of genetic algorithm. So, we start with any population so we can say that initial population randomly we select some population actually and then there is a concept call the converge. That means, if we say there is no improvement in the next population then we can say that stop it here no more progress, but if we say there is some possibility of progressing.

So, then we can go and then selection it basically out of these populations select the best and those best basically responsive for reproduction if generates the next population. So, it will go there and again start there with the next population is converge, converge mean we can achieve our goal or not that mean complete or not. So, this way this algorithm is basically is a process you continuous or iterative process which run for long until we can come to the converge. So, converge means in the sense that we can search the optimum result. So, that is the converge now here many things are hidden.

So, how this population is related to our problem solving, and how the selection and reproduction can be realised. So, that from one population the another population can we obtained. So, the another population means it is basically towards the better solution and these are the process selection and reproduction is basically is a fundamental block for probabilistic searching. So, we will discuss about this concept in details in the next class so fine.

So, this is the basic framework of the genetic algorithm and based on this basic framework there are many other frameworks also have been proposed, all these things we will discuss in the next class.

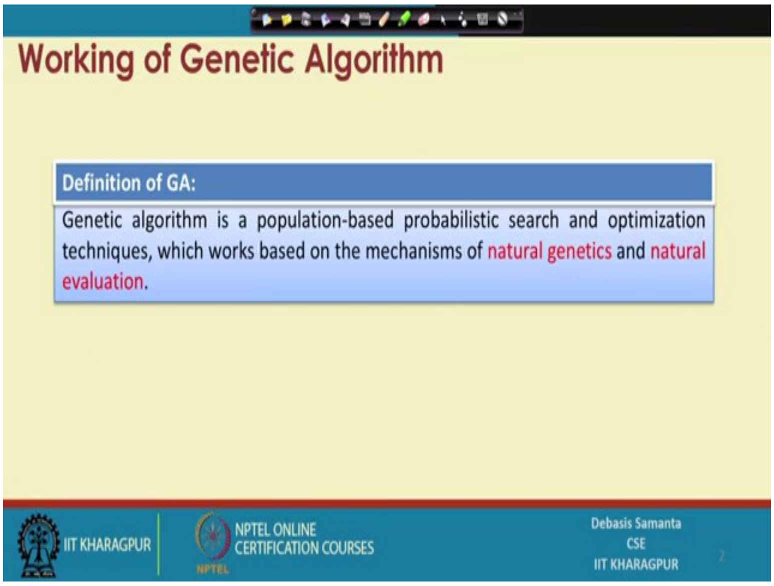
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 15
Concept of Genetic Algorithm (Contd.) and GA Strategies

So, so we are discussing about genetic algorithm, and we have discussed about that two biological process namely genetics and evolution, how they can be used to solve optimization problem in the form of evolutionary algorithm called the genetic algorithm. And based on this concept, we can define the GA as a population based random search, this means from one population to another population we have to search for the best solution.

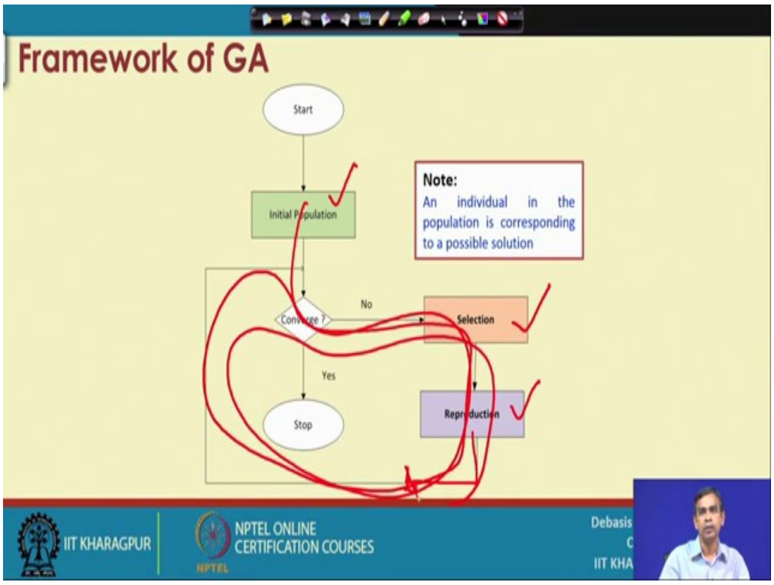
(Refer Slide Time: 00:53)



So, it is called the population based probabilistic search, and that is used to solve optimization problem, and such a population-based probability search is based on the concept of natural genetics, and then natural evolution.

So, today we will discuss about details about it. So, what does a population means, and then the random search how it can be carried out by means of the genetics, and then natural selection or evolution.

(Refer Slide Time: 01:27)



So, the algorithm follows a framework which is shown here, and in this framework, we see three basic building blocks. The first is the called the population, and then is the selection, and then reproduction, and then these are the basically control to process I mean towards this one. So, this basically eventually is a searching process and is a probabilistic searching process actually. Now, here few things are important the population, the selection, and then reproduction. So, the selection is basically follows the concept of evolution proposed by Charles Darwin, and reproduction is the concept it is basically the concept of genetics proposed by Gregory Johan mandala.

So, they basically the idea that is their idea is basically how from a current population initially it is initial population the process can be moved. So, that it can gives a new population here, and then it can iterate the same thing until we can have the search complete. This is called the converge. And finally, we get the solution it is called the optimum solution. So, essentially it is basically starting to this one we have to obtain from here, but by means of this process this process so long the conversation is complete and we can get it.

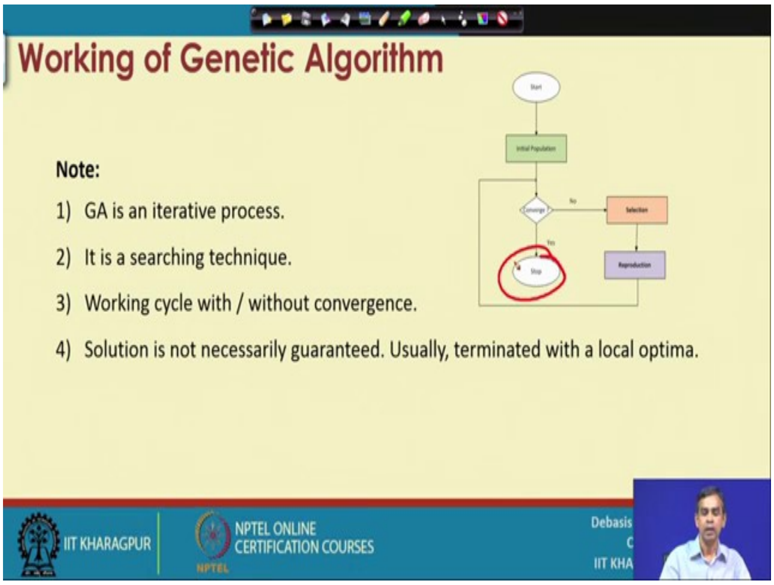
So, the idea it is like this, and this kind of concept it is basically the genetic algorithm concept, and this is the basic framework of the GA. And here one thing that population here is basically is an individual, and this population can be linked to is a solution. Or in other words, a population or an individual belongs to a population basically is a possible solution. Now, this is very important and individual in this population is basically a possible solution, these three things are very important.

So, somehow if we can represent a possible solution, by means of an individual, and a number of individual is selected and then it gives a generation that is called a population. And from this generation if we follow this selection and reproduction operation so, that we can get another population or another generation then, and this another generation is significant in terms of the improved solution.

So, if it gives a solution and it gives another population, who is consists of improved solution, then if we iterate this one and then by this iteration iterative process, we can get from one solution to more improved solution, and we can stop this searching if we satisfy some convergence criteria. That means, when we have to stop this seemingly is basically the infinite loop.

So, this process is basically followed in genetic algorithm and to understand the genetic algorithm we have to understand, how this initial population can be generated. That means, a possible solution can be converted to an individual, and then how the selection, and then how reproduction operation can be carried out, or can be realized. We will learn all these things in the subsequent lectures one by one. First we will discussed about this framework the framework of GA and it is variation. So, in this lecture we will limit our discussion to that only it will help us to understand the concept further, and then later on we will discuss the other concept.

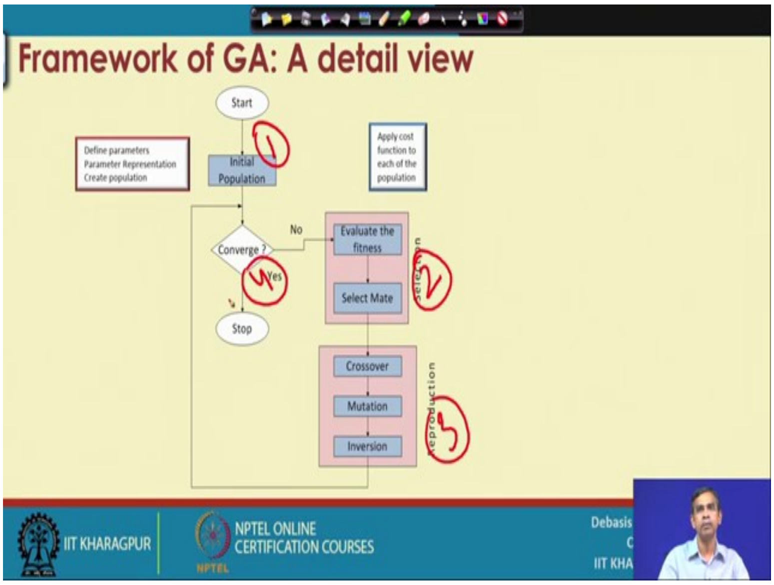
(Refer Slide Time: 05:46)



So, let us proceed here. So, as we have learnt it the genetic algorithm works as an iterative process, and in this iteration it basically pursue the searching. Searching for the base solution, and this working cycle is basically continued with certain convergence criteria. That means, whether or you have to continue the search, or you have to stop it. Now, one important thing is that solution that it ultimately gives here, the solution not necessarily to be guaranteed, but it can give near optimum solution. So, near optimum solution is sometime called a local optima, and then guaranteed optimum solution is called the global optimum.

So, there are minima as I told you and then out of this minimum, the global minimum is the guaranteed solution. But sometimes genetic algorithm can converse into a local optima, and that local optima may be sufficient to find the optimum solution, which cannot be solved using traditional approach to find this one. So, the genetic algorithm essential does not give you the correct solution always, but is a near correct solution actually.

(Refer Slide Time: 07:02)



Now this is the concept of the genetic algorithm that is followed their, And I just want to detail as it the different steps involved, namely the solution generation, and then the selection operation, the reproduction operation, and all this thing that is there in this framework.

So, we have to start it, whenever we start with; that means, we have certain random solution at our hand. So, the random solution basically gives an initial population. Now in order to have this random solution, we have to have full idea about the different parameters which are involved in this optimization problem, and then all these parameters are to be taken into account and this should be represented in some form for each the GA framework can handle them so, it is called the parameter representation, once the parameters are identified and then parameters are represented precisely, then with the help of this representation we can generate the population, population means random solution.

So, initially if we generate a population such a population is called initial population. Once the initial population is observed we have to check about whether the initial population already got the optimum result or not, that means, the best solution or not. If it is yes, then definitely we can stop this process; that means, our work is done. If it is not then we have to go to the next population generation. Now so far, the next population generation is concerned it basically needs two tasks, these tasks are relative selection and in reproduction. Now so far, these two tasks is concerned it basically evaluate the fitness of a solution. That means, how solution is good so, that is basically evaluation and now whenever this evolution is there so, it basically considered some cost function, we will learn about that how to evaluate a solution or how to evaluate an individual in the population so, that it fitness value can be calculated.

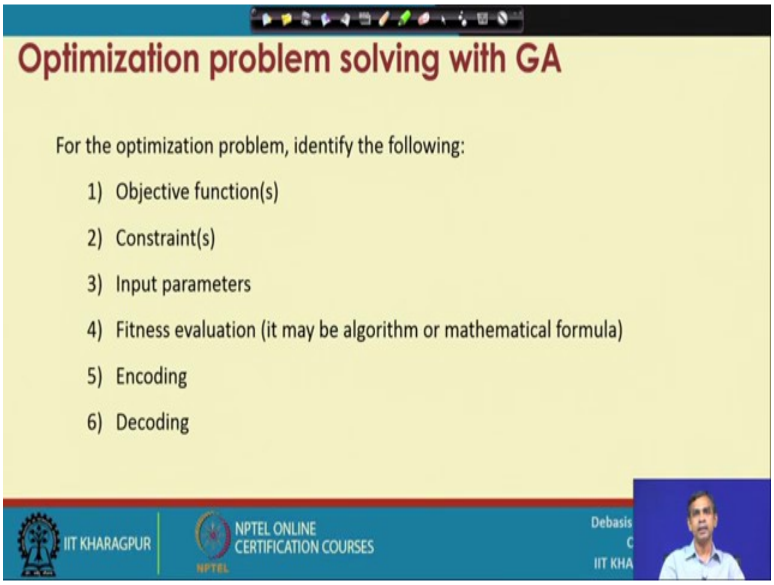
So, these are the theory are there we will follow the theory, and then based on the evaluate, we have to select mating pool. That means, out of the individual who can be responsible for the mating process. So, that they can produce next offspring. So, that there is a select mating there is an again there is a theorem by which from this fitness evaluation value we can select the mate. Once the mating is done; that means, the mating pool is created; that means, this individual will mate with other individual and so on, so on. So, this mating pool will undergo reproduction scheme. Now reproduction is a there are some few steps the reproduction crossover, reproduction by means of mutation, reproduction by means of inversion.

So, this reproduction in fact, produced from a mating pair to another individual. So, successfully there are set of mating pairs from these mating pairs, we can obtain one or more results or it is called the offspring. So, these offspring produced the next generation. So, next generation will be here. Again the next generation will be tested, whether the next generation achieved the best result or not, if pairs will stop it here. If no, we can repeat the same procedure, until this convergence criteria is successful.

So, in genetic algorithm few things are important how to create a new population. And then how the selection can be done. Selection by means of evaluation and then mating pool generation, and once it is there then how to go do the reproduction. That means, how the crossover mutation and inversion can be carried out so, that next population can be obtained, and for each population we have to check the convergence.

So, this is the one important tasks, this is second, this is third and finally, convergence. So, learning a genetic algorithm in fact, learning these four tasks in details. So, we will learn all these four tasks in details and then finally, we will see how given an optimization problem, we can solve this problem using genetic algorithm.

(Refer Slide Time: 11:40)



Now, will discuss about few things here. For the optimization problem, we have to consider few things, first given an optimization problem means we have given an objective function. Objective function in fact, defined in terms of some input parameters, basically with the these are the parameters whose values decides the value of the objective function. And as I mentioned already an optimization problem is specified by means of objective function, and then a constraint set of constraints.

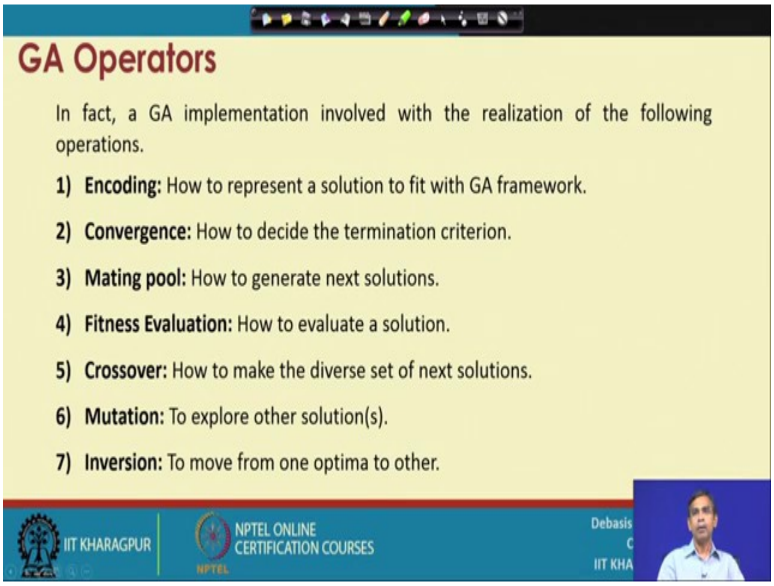
The constraints are basically the requirement by which all these values should satisfy so, that the optimization function will get it is value. And so, input parameters are involved there so, input parameters are basically the input values to the system, and then the fitness evaluation and for every solution we have to calculate some fitness value. That means, if say suppose solution is the, optimum solution one or global solution one, then it should have the highest fitness value. If it is very far from the global optimum then it fitness value is also very far from the optimum fitness values.

Now in order to represents a solution in the form of a genetic algorithms individual, then we have to follow encoding. Encoding means how the parameters can be represented so, that it leads to a chromosome. Chromosome is the basic concept that is there in the genetic.

So, encoding is nothing but representation of chromosome for a solution, and one solution we can consider an individual. So, chromosome is basically decide an individual then. Now chromosome is basically encoded form; that means some symbolic representation. So, we have to follow decoding so, that from this symbolic representation we can get the actual values. In other words, a parameters that is there we have to encode it so, that the chromosome can be represented. This chromosome defined the individuals proper solution or population, and for many solutions the many chromosomes and then many what is called the individuals, and then population is generation.

Now encoding is basically a process by which a chromosome can be obtained for a given problems or in terms of input parameters, how the encoding can be done so, that input parameters can be converted into the encoded form, and decoding is basically the reverse form of the encoding.

(Refer Slide Time: 14:33)

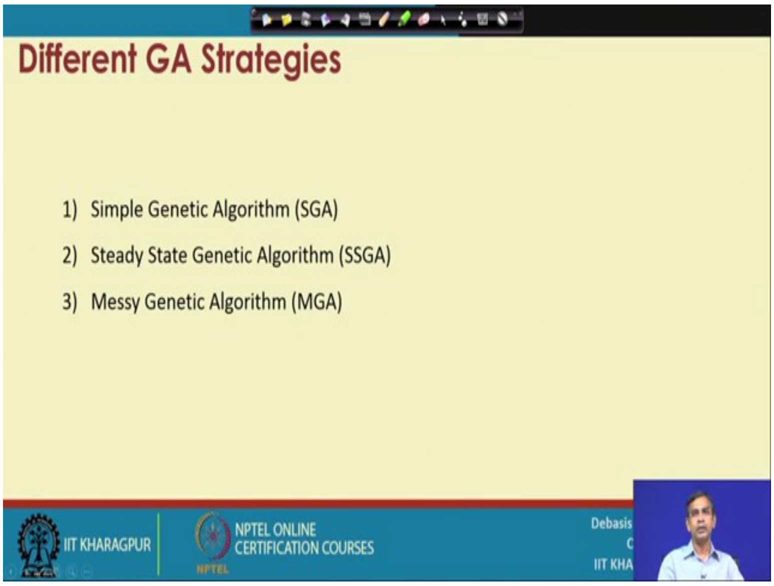


Now, so, these are the optimization problem solving approach with GA, and so, far the GA operation is concerned, those are the operation that we have mentioned is carried out by means of called the operators, those operators are basically the functions like.

So, for example, encoding convergence, mating pool, fitness evolution, crossover, mutation inversion these are called operators. These are basically the functions like so, if we give an input this function will produce an output; that means, if encoding is concerned it basically represents a solution or that chromosome for a given input parameter or set of parameters. Now convergence is basically if we get a population or a generation, it will check whether we have reached to the termination condition or not, then you have to stop it or you have to continue the process of the genetic algorithm. Then mating pool is basically given a population we have to create the mating pool. So, mating pool is an function or it is an operator, by which input is a population and it produced the output as the pools, mating pool. Similarly, fitness evolution it is also a function input to this function is basically an individual it will return a result giving the fitness value of that individual or solution.

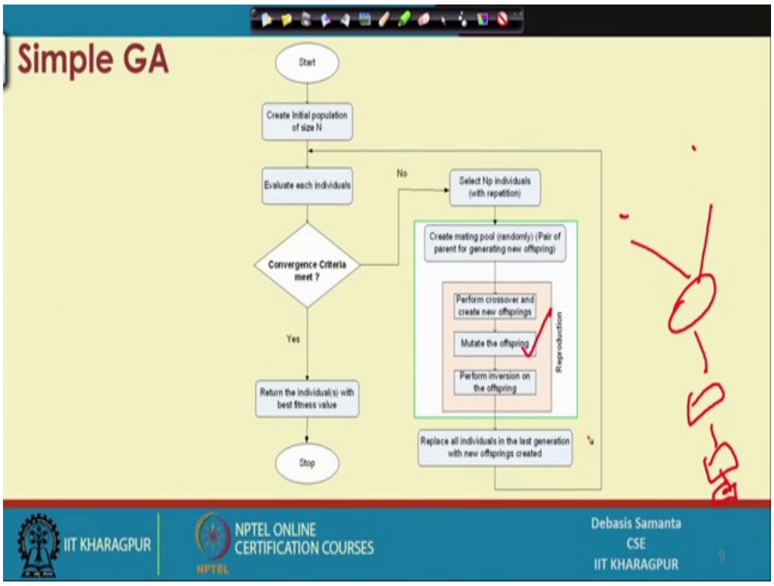
Crossover is basically if we pass two chromosomes to this operator, it will produce two or more offspring. So, basically the crossover is an operator, inversion is basically to jump from one optimum value to another optimum value, how it can be done. So, if we give a population here it will produce another new population that can leads to a better optimum value like. So, these are the operators. So, learning GA is basically learning all these operators. So, we will discuss about how these operators can be realized in terms of some simple problem.

(Refer Slide Time: 16:33)



Now, before going to actual learning about the different operators, there are few what is called the strategies for the GA. One is called simple GA, it is also called simple genetic algorithm, and another is called the steady state GA or steady genetic the algorithm, and alternative to these two GA strategies there is also one called messy genetic algorithm. In this lecture we will only discuss our we limit our discussion to the first two than SGA and SSGA, simple genetic algorithm and steady state genetic algorithm these are the most widely followed strategies. So, for the genetic algorithm is concerned once we learn these two strategies will be able to learn the genetic algorithm later on who in terms of their different operators.

(Refer Slide Time: 17:20)



So, let us first start with the simple genetic algorithm, it is called the SGA. And this architecture this block diagram or flow chart, shows how the simple genetic algorithm works. We may note that this simple genetic algorithm is basically a little bit detailization of the genetic algorithm framework that we have already learned now. That means, the different researchers followed the different way to solve the genetic algorithm, they just do certain different follow certain different strategies.

So, we will discuss about the strategies that is followed, they are in simple genetic algorithm now we will just step by step here. So, first as the start is; obviously, the starting point of any process, and then the initial population creation. Now, here one parameter called the GA parameter first followed. So, initial population means it is a collection of solutions. So, random solutions now whenever we say the collection of random solution then what is the size. So, it is a programmer who decides the size, let N be the number, N may be some times hundred, some user may follow N as a thousand so; obviously, if we follow large values of N may be that we can come into the quick the solution or better solution, but at the cost of more timing.

So, if we it is take the population size less. That means, value of N is small then we may come quickly terminate the solution, but it may not give you the correct result always. So, as a trade off course, there, in this strategy population size is an important parameter. So, simple GA, GA consider a good value of N, the N is the first GA parameter the genetic algorithm parameter.

Now once the population, population's size is decided, and following some procedure of creating the random solution as the initial population, the next task in according the simple GA is basically evaluating each individual. So, it is called the evaluation that evolution means the fitness evaluation. So, it basically for each an individual solution in this current population, the operator or function evaluate the fitness value. So, that is called the evolution.

So, this is a second step and then after the evolution of individual solutions have known, it basically come to the convergence criteria checking. For each individual it will check that, whether see suppose we know that the highest fitness value is possible for a solution is this one. If any individual has this highest fitness value, then we can say that we can achieve the convergence criteria. So, we can stop it there. Now if convergence criteria satisfy we can say yes and return the individuals with the base fitness value and that is the solution.

So, these basically the solution now if these convergence criteria are not satisfied, then we have to go to the next step. The next step is basically select N_p the another parameter in simple GA, N_p denotes the subset of this value of n, N maybe say 500, and N_p maybe say 20% of 500, maybe say 100 so, select 100 individuals from these individuals. So, this is subset selection, and these basically allow with repetition.

Now whenever it is repetition we can select here basically how we can select. So, you the repetition is allowed; that means, we can select at random. So, we select one returned to this individual again, select returned the same individual and so on so, if we proceed if we follow the same procedure for N_p times. So, it will give N_p individual selection, from this population with repetition.

So, this is the idea and we can select it at a random fashion. Randomly we can choose one and then select and then being into the in this pool then next again this one and so on so on. So, N_p individuals with this selected, and with repetition; that means, same individual may be selected one or more times here. So, this algorithm allows this one to select one individual more than one type.

Now, this select again there are certain heuristics or some principle or policy to be followed. We will discuss about what the policy that needs to be followed in order to realize the selection operation. Next once the N_p individuals a subset of this set is selected. Our next procedure is this is basically the selection algorithm that is there. So, selection operation and I told you that selection operation is based on the concept of evolution, which basically has the four premises, heredity, diversity, and then ranking and then selection. Anyway those things will be discussed again I will discuss the selection procedure in details, and next whenever the select N_p individuals are selected there create mating pool, and here again mating pool should be created randomly. So, these are the different individual if it is they are.

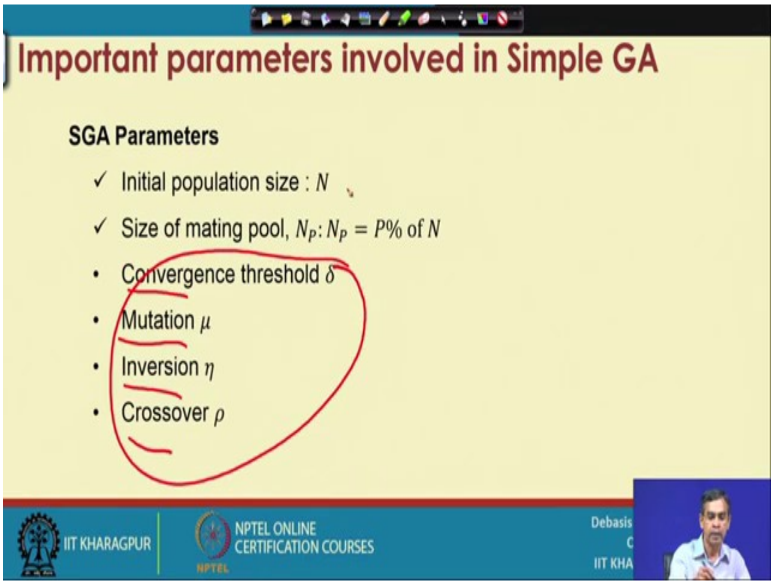
So, randomly this one or this, one mating pool. So, this one and this one are mating pool. This one this one another mating pool, this one this one mating pool. So, this way you can randomly select certain pairs, and these pairs gives you the mating pool. So, you can create mating pool is basically random process. So, this is also random process, this is also random process. Here also initial population generation random process, that is genetic algorithm we call the probabilistic search, random search.

Now once the random randomly those mating pool is selected then for each individual there is a chromosome. So, from this they are haploid and then from a haploid. So, cross over and then diploid it will be created. So, this basically concept the crossover it is haploid here and in this algorithm and then reproduction is there, and mutation is basically is a another approach. It is also in nature say that all of a sudden sub chromosome is there, there is a some breaking in the code or DNA code, like or genetic code. So, all of a sudden drastic change or some changes in the genetic code, it basically called the mutation, and inversion is basically I mean if mutation is a very minor changes where inversion is a very detailed changes. So, the detail changes are called inversion.

So, so what will happen is. So, if the two I mean chromosomes from their offspring is created, this offspring undergo mutation, it is not necessary mutation should go or not go with certain probability, and then inversion then finally, it will give you the new offspring. So, this new offspring will be stored into the new population or new generation, and then all these new generations that has been obtained, then these basically out of these N_p . So, certain number of new offspring will be generated.

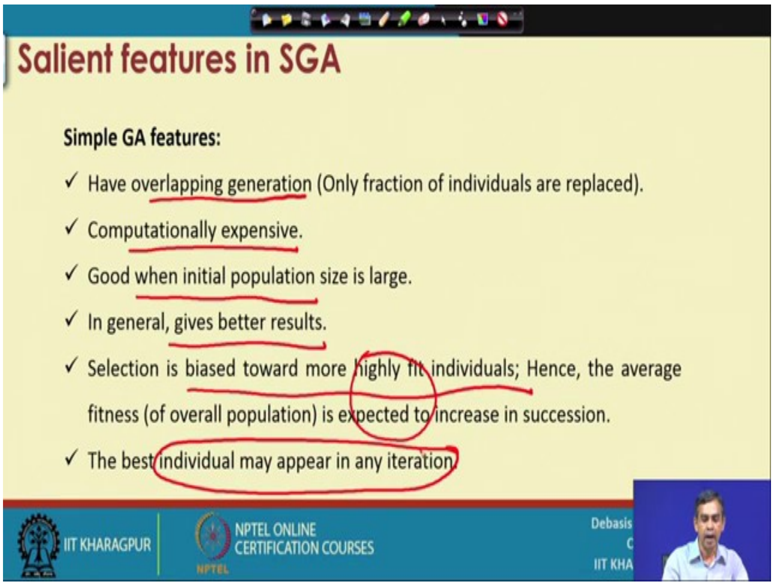
So, here according this simple GA replace all these new individual that obtained, in the last generation with new offspring created. So, we have selected out of in N_p those N_p will be replaced by the new offspring. So, it will remain the same size. So, here basically N and then N_p . So, these N_p will produce new N_p , and then new N_p will be replaced by the old N_p . So, the population size will be N and then new population the next generation will be there. So, this is the idea of the simple GA.

(Refer Slide Time: 25:28)



And so, next idea is other than simple GA it is called the, so far, the simple GA is concerned few things are important. So, as I told you in the simple GA few parameters N these are initial size of the population N_p is basically, size of the mating pool it basically p of n , p may be some value decided by the programmer, all these N_p these are the value decided by the programmer, who will use these genetic algorithm to solve their problem. And other than these parameters few other parameters are also involved these are a convergent threshold; that means, it is basically the range by which we can take so, that our result is near optimal or near minimum or maximum, and then few other parameters are there so, these are called mutation parameters inversion parameters and crossing over parameters. So, we will discuss about all these parameters when we will discuss all these operations one by one. So, till time we can keep it on hold.

(Refer Slide Time: 26:25)

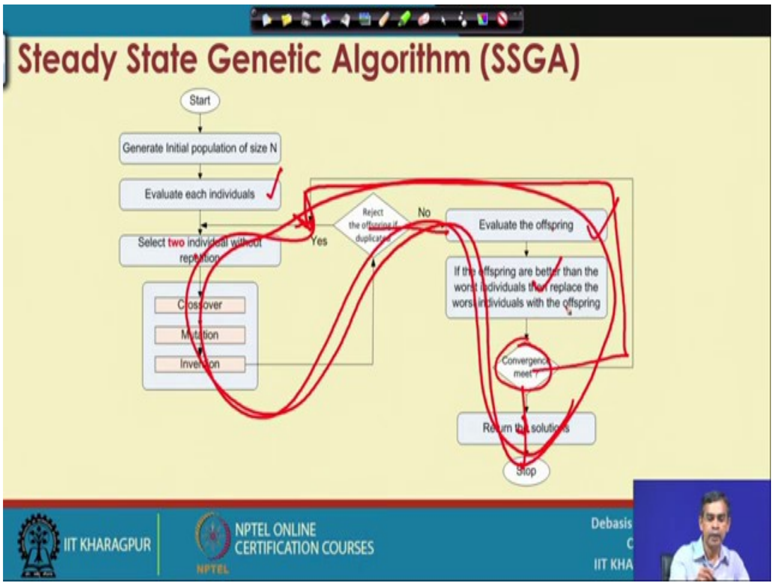


Now, let us see the other algorithm that is there. So, to discuss about it now there are few important features in SGA means in which situation we should follow the simple genetic algorithm so, simple genetic algorithm always produces overlapping generation; that means, only fraction of individuals is replaced. That means, from the current generation to the next generation, few individuals are remains common. So, it is called the overlapping generation. So, these the one characteristics because, some solutions are common in between the two-successive generation.

And this simple GA is usually compression expressing compared to the other GA strategies, and this kind of GA is applicable, when initial population size is large. This general usually gives better results compared to the other strategies, and here in this algorithm selection is biased towards the more highly fit individuals. That means, it basically selects those individual which are having the highest fitness value. But it is observed that sometimes selecting, non-highest fit individual value and then taking the crossover can lead to a better solution. So, this has the one what is called the issues as here. So, that highly fit individual selection not always useful rather, if we select some other individual having the lower fitness value may give the better result or may give quick result.

Anyway so, these are the another thing and these are the best individual may appear in any iteration. That means, it can give the best result in any iteration, and then that can be one achievement or that good point so, that we can terminate quickly if you are lucky enough then with one iteration maybe you can terminate, but if you are not then you have to repeat it again. But the chance that it will terminate very quickly higher compared to the other strategies.

(Refer Slide Time: 28:30).



Now so far, the other strategy is concerned, we will discuss about next strategy and then we will decide we can understand about what are the difference between SGA versus this strategy, now the next strategy is called steady state genetic algorithm. Now let us understand what are the strategy it is, as in the simple genetic algorithm it will start with the initial population size N, N is decided by the programmer evaluate each individual that is the fitness evaluation. Now here there are actually N_p number of selection in case of SGA, but this SSG only select two individuals, now these two individuals one selected they then go for the reproduction. That means, we do not have to do the mating pool creation here unlike in case of SGA known it.

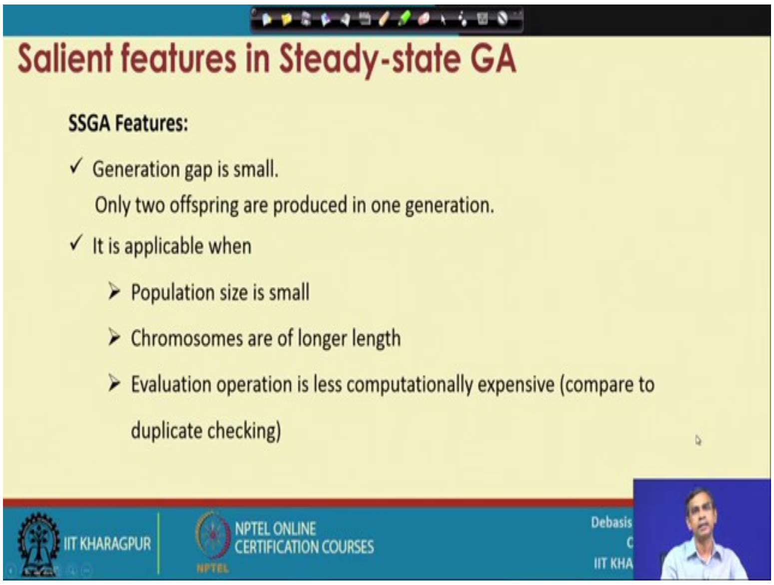
So, from the two individuals will be selected from the current population and then they will produce the offspring. Now here again go to that reject the offspring it is duplicate. Now the offspring that will be produced if we see it is already there in this population, then we should reject this offspring then we can repeat another individual selection from here and then another offspring can be created. So, if it creates a new offspring which is not repeated or not available already there then we can go if it is no. So, evaluate the offspring. That means, the fitness value of the offspring will be created, but in this case only one evolution is required, now here after the fitness value is created if the offspring are better than the worst individual.

So, in this population there are few individuals whose fitness value is worst. So, if the current offspring fitness value is improved value, that is the fitness value is greater than the any worst individual fitness value, then we should replace these worst individuals by these new individuals new offspring. That means, it in each iteration it will replace one offspring with worst witness value by a better witness value.

After replacing it will produce the next generation, but in the next generation you can see only one offspring is different anyway. So, if the next generation satisfy the convergence criteria, then we can stop and then these are the solutions. So, here you can see the solution having the highest fitness value is the ultimate solution. If it is it does not satisfy the convergence criteria you can go it, again proceed the same procedure select two individuals. So, it is basically the root. So, the iteration will be there.

Now we can understand the difference between SGA and SSGA. SSGA always selects in each iteration or changes the population with a larger gap; that means, gap is at least by N_p , but in this case the gap is very small, only the 2 successive population is differed by means of only one two solutions.

(Refer Slide Time: 31:41)



So, this is the fundamental difference that we can easily understand, and here is the some features so, far the SSGA is concerned as I told you the generation gap is small only two offspring's are produced in one generation, it is applicable usually when the population size is small, chromosomes are of longer weak, and evolution operation is less computationally expensive; that means, when the evolution is too much expensive to calculate the fitness value of an individual, then we can follow it and chromosome length is basically if the solution needs a large number of parameters then usually we can follow this technique.

So, we have learned about SSGA, and limitation in SSGA is that compared to the SGA of course, there is a chance of stuck at local optima. If crossover mutation inversion is not followed properly, premature convergence usually occurs which is not occur generally in case of SGA, and it is susceptible stagnation. That means, inferiors are neglected or removed and keeps making more trials for very long period of time. So, sometimes it is observed that if the inferior individuals are taking into the mating, they can lead to the better solution so, but it is ignored here. So, it may come to a stagnation situation.

Okay. So, we have learned about SGA and then SSGA, the two strategies and based on these two strategies how the genetic algorithm can work, and then in order to understand the working of genetic algorithm, we have to study about the different operators. So, in our subsequent lectures we will learn about the different operators that is here, namely encoding, the crossover, mutation, the selection fitness evaluation and all these things ok.

Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

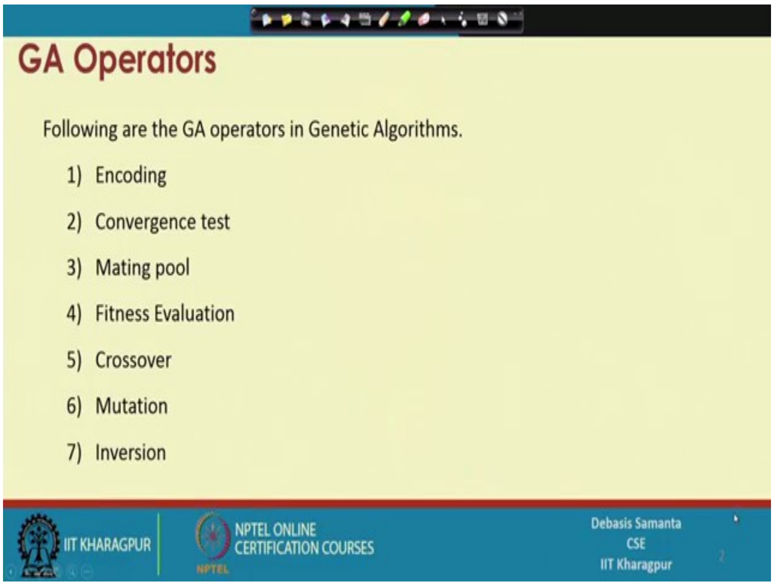
Lecture - 16
GA Operator: Encoding schemes

We have learned the basic architecture of genetic algorithm. And in the basic architecture, there are many operations are involved. One operation is creating the initial population and then convergence testing and then selection operation and finally, the reproduction operation. Reproduction include cross over mutation and inversion.

Now, so far, the first operation namely, how to create initial population. It basically require to learn about the encoding scheme. Encoding scheme implies how a problem can be encoded. So that the GA architecture can use it and then follow it is operation to produce the output result.

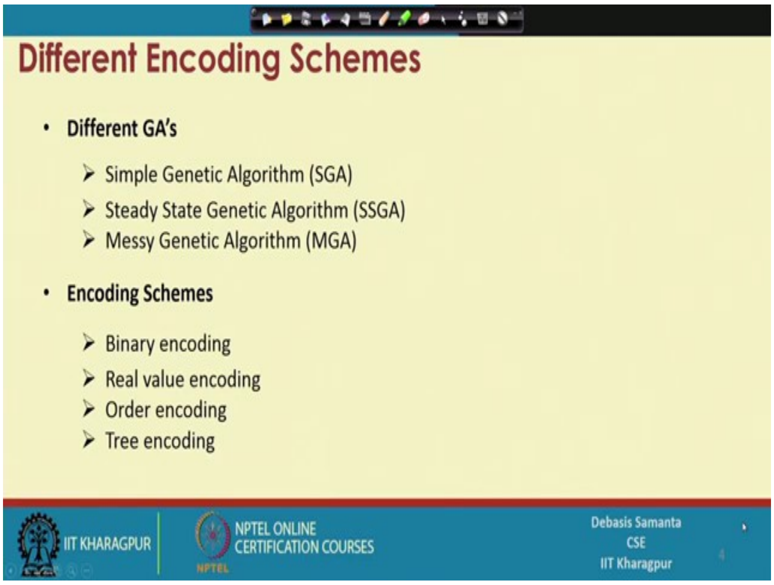
Today, we will discuss about, in this direction the encoding operation.

(Refer Slide Time: 01:11)



Now, different operations again I repeat it the encoding, the convergence testing then creating mating pool fitness evaluation it is basically part of the selection and then. So far the reproduction is called the operation or crossover mutation and inversion.

(Refer Slide Time: 01:33)

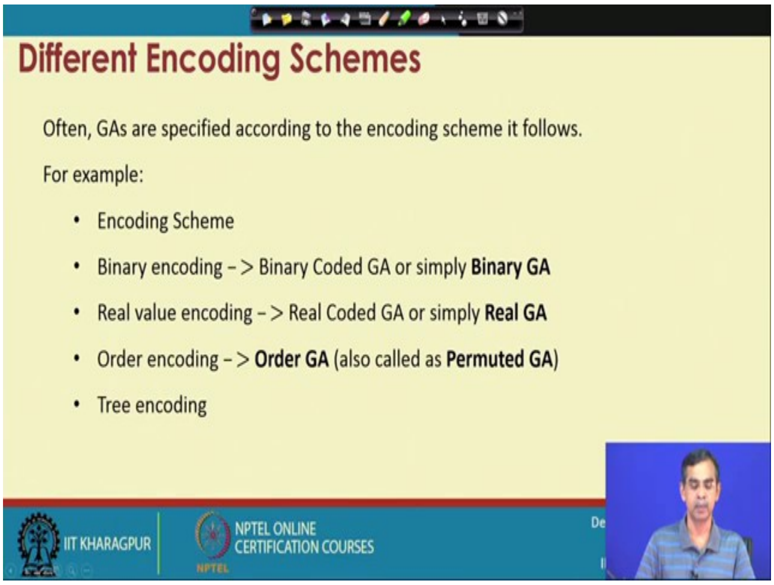


Today, we will discuss about encoding and fine different genetic algorithm we have discussed about a brief about the different genetic algorithm like say simple genetic algorithm and steady state genetic algorithm and also messy genetic algorithm. In fact, they are different from the point of view how the different operations can be carried out. And then also there is one important difference between among this is basically what are the different encoding scheme that they can follow.

For example, for the simple genetic algorithm and steady state genetic algorithms are concerned. They follow the constant length encoding, where is the messy genetic algorithm constant the variable length encoding. That means, anyway let see exactly what are the different encoding schemes are there. Then we will be able to understand about the difference between messy genetic algorithm then others.

Anyway, for the different encoding schemes are concerned, we have listed here few important encoding scheme which are most popular here. The first is binary encoding and then real value encoding, order encoding and then tree encoding. All these encoding scheme will be covered one by one.

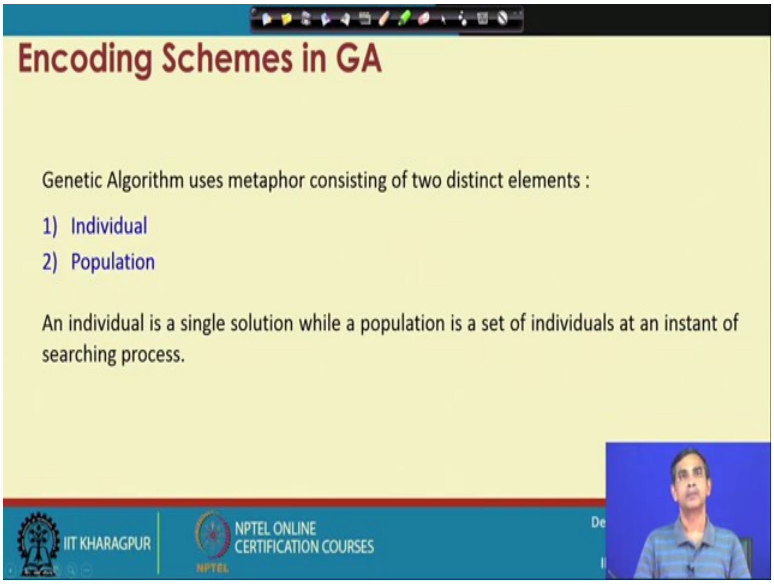
(Refer Slide Time: 02:43)



First, we will start about the binary encoding and before going to this if a particular genetic algorithm follow a particular encoding scheme, then according to the encoding scheme that this algorithm follows it is termed as that GA. For example, if your genetic algorithm follows the binary encoding to create the population, then it is called the binary GA.

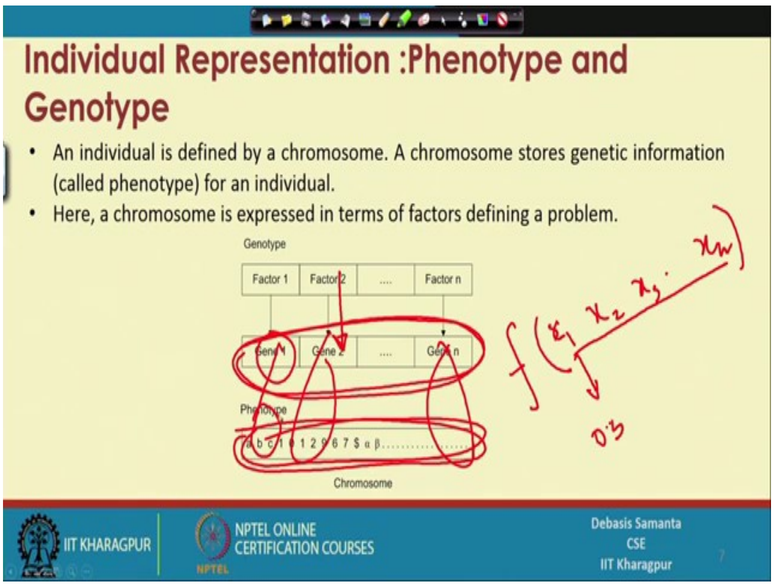
The real value encoding is another approach if it is followed then the GA is called real GA. Order encoding, if it is followed in genetic algorithm then it is called order GA. Sometimes order GA is also called permitted GA. And then if it is follow tree encoding mechanism then it is called the tree encoded GA.

(Refer Slide Time: 03:27)



Now, encoding scheme, for the encoding scheme in genetic algorithm is concerned. Basically, there are two different things involved in genetic algorithm architecture, one is individual and another is population. Individual basically related to a solution a possible solution or a prospective solution and a set of prospective solution, if it is included then it is called a population. Population is basically is a set of individuals and individual is a particular solution and population usually at any instant. What are the different at any instant of the searching process searching for the best solution. In fact, at any instant of the searching process the set of solutions are basically the population of that instant.

(Refer Slide Time: 04:23)



Now, let us first discuss about before going to the encoding scheme the two concepts in order to encode a particular solution, it is called the phenotype and the genotype. Now, as I told you earlier, the genetic algorithm follows the concept of genetics. And in the concept of genetics, the chromosomes play an important role. A chromosome is basically is a collection of genes and a particular genes gives a particular DNA for an individual. Every individual has it is own DNA code that is the gene combination.

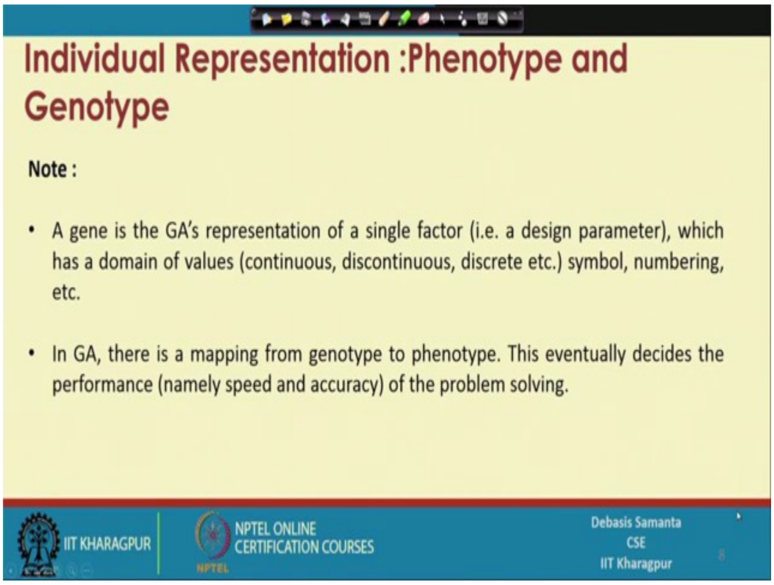
Now, the basic structure of all these chromosomes is called the genotype. Basic structure means, as you know genotypes is basically the different from one problem to another. That means, for your problem you have to decide exactly the genetics genotype.

What is the genotype? I can give an idea. Suppose you have to your objective function consist of a different variable like $x_1, x_2, x_3, \wedge x_n$ so. That means, we have to find these are the basically input parameters and we have to optimise one function which is f for given values of these parameters. Now, here $x_1, x_2, x_3, \wedge x_n$ are the n number of design parameters. It is also called the factors. For example, Factor 1 is x_1, Factor 2 is x_2, and Factor n is x_n.

Now, if x_1 has a typical value say 0.5 then it is called the gene value for the parameter x_1. Similarly, for every factor there is a gene value that mean the value of this parameter at that instant. These combined is called the gene values for the different factors or the different design parameters in the problem. At any instant, the values of all the factors constitute what is called the phenotype. That means, gene 1 this part, gene 2 is this part and gene n this part is constitute the entire what is called the things. It is called the phenotype.

Genotype and phenotype is involved in this way. And so, encoding is basically how a factor can be encoded to give some value. That means, gene value. We will see the different method for the encoding scheme.

(Refer Slide Time: 07:10)

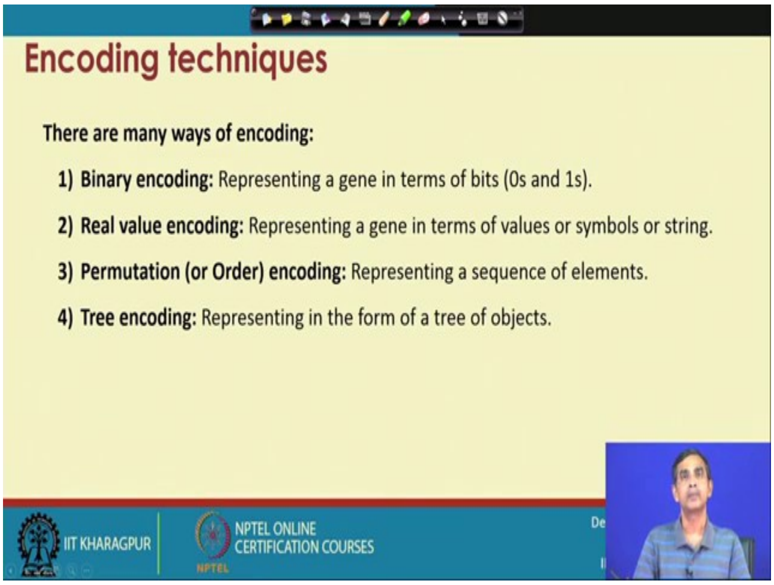


Fine, before going to understand about the binary encoding. We can understand or we can say that a gene is the GA's representation of a unit, that is a single factor that is a design parameter.

The design parameter may have different values. They can be defined in the discrete domain; they can be of continuous or some symbolic values or number etcetera. That mean gene value can be anything and that is according to the requirement of the problem that you are going to solve.

In GA, in fact, there is a need to mapping from genotype to phenotype. That mean for each factors or design parameters that is there in the phenotype. How to map to some values of it. That is the genotype this eventually decides the performance of the algorithm. That means, the efficiency and accuracy of your problem solving. If you design if you design the encoding scheme effectively and properly, then you can get the result at the earliest and also the correct result.

(Refer Slide Time: 08:35)



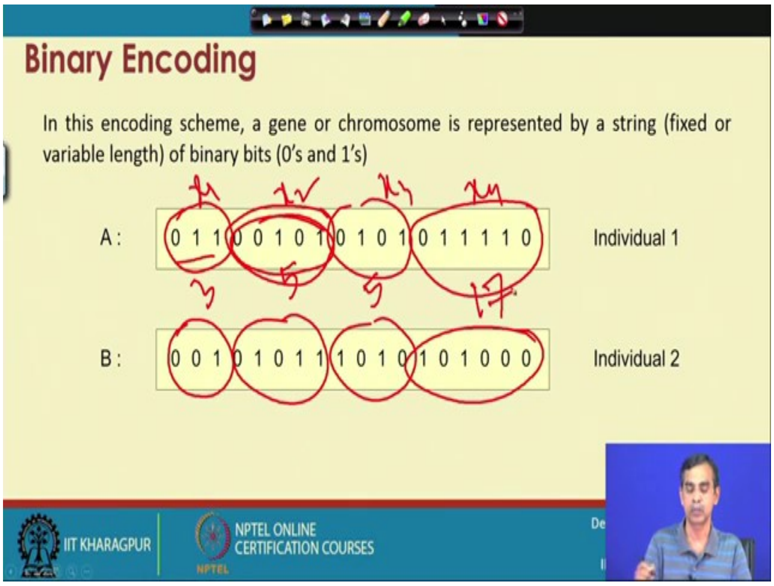
Now, we will come to the different encoding things techniques. I have already told four different techniques. The binary encoding, the real value encoding and then order encoding or permuted encoding and tree encoding. Now, binary encoding as the name implies it follows the gene value in terms of 0s and 1s. That means, the binary representation. You know anything whether it is a number it is a real number; it is a symbol or it is some other representation can be represented using binary encoding scheme.

It basically gives a gene value in terms of only two bits called 0s and 1s. On the other hand, real value encoding scheme is very convenient to understand from a programmer point of view. It is basically straight way to an exactly what is the value for a particular parameter is to be stored there. That is why real value encoding. It is if you want to I mean say factor is a name. The name can be can consist of 20 characters alpha numeric whatever it is that you can store like that.

Now, order GA is a special, special case of encoding. It requires not to solve all program, but there are some programs where the sequence of elements matters. If it is a problem like this, then we can encode that kind of problem using the order encoding scheme.

And tree encoding scheme is a special form of the encoding mechanism where it is stored in the form of a tree. T is a concept it is a structure with which we can solve we can represent many what is called the problem and we will learn about tree encoding with an example that understand it.

(Refer Slide Time: 10:32)



Now, I will first start with binary encoding. As I told you in this encoding scheme a gene the collection of genes basically constitutes a chromosome. A gene is represented by a string of 0's and 1's is a binary string. Now if we follow the fixed length of the chromosome, then they are basically SCA or SSJ.

On the other hand, if it is a variable length chromosome then it is called the messy GA. Now, first let us see what is the length of the chromosome. Now, in this example if we see the number of 0's and 1's altogether is 18th. The length of the chromosome is 18th.

Now, here for example, another this is the length of the chromosome. Length means this is the length of the chromosome and this is the solution 1; that means, 1 individual and this is the individual 2. 2 chromosomes are represented here we have name this chromosome as A and B or Individual 1 and Individual 2.

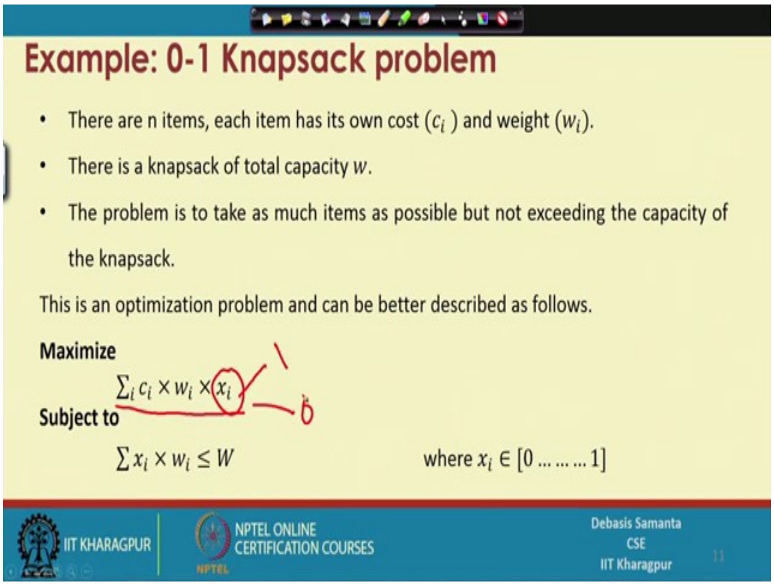
Now, here if this chromosome can be like this we can think about it. Say, these are the first 3 bits for one factor and these are the 5 bits for another factors and then this is the 4 bits for another factor and then finally, these are the 9 bits to another factor. We can say this is the parameter $x_1, x_2, x_3, \wedge x_4$. Likewise, this is another gene for another factors representing these value.

Here, we can see the four factors or four design parameters have been encoded with their binary value. Now, if we say the binary value this represents the four. For the x_1 is concerned and these represents this is basically 3 right and this is 5 and so on. Basically, the binary equivalent a decimal equivalent of this binary representation it basically. It is a 3, 5 and this is also 5 and this is basically 2, 4, 6, 8. 8, 6, 14 to 16 and 17. These basically represent 17 and so on.

These are different value that has been represented by these combinations. Whole the things constitute one what is called the chromosomes. This is an essential thing in the binary encoding. Basically, each design parameter can be represented by means of some binary string. And as you know, binary encoding is powerful to represent any value whether it is an integer. Integer can be coded using decimal to binary conversion formula. Real value also can be because there is also formulation by which any real number can be converted in the binary one and then any symbolic representation any characters also can be converted; a string of character also can be converted.

Anything can be converted using the binary representation of the gene. So, the binary representations follow in a binary encoding and this is the idea, I just want to give one example, so that you can understand about the application of binary encoding to solve some optimization problem.

(Refer Slide Time: 14:04)



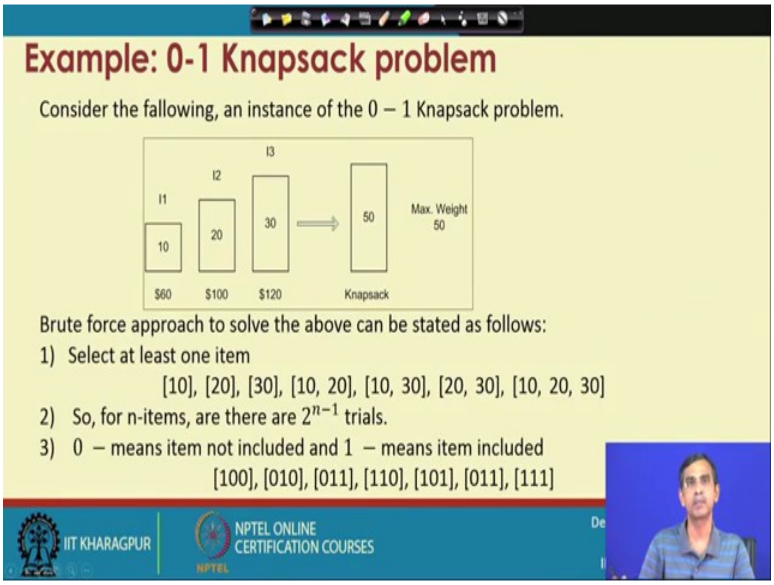
I will discuss one optimization problem it is called Knapsack problem and more precisely it is called 0- 1 Knapsack problem why the name 0- 1 we will understand about it.

Now, in this problem basically there are n items given to you and each items are specified by a by its cost and then weight. N items and then N items has their cost and weights known to you. And there is a Knapsack and then total capacity of Knapsack also given to you.

Now, here the problem is to take as much as items as possible, but not exceeding the capacity of the Knapsack we have collected. As an example, suppose a thief enter into a showroom with a knapsack in his and knapsack his own capacity and the different cost of the different items are known there, and the thief has to collect the maximum elements from there, that he can put all the elements into his knapsack. And so that he can also I mean gain maximum I mean. For the thief is concerned he can take the maximum amount from there. Basically, within the limited weight he has to collect the items and each items having their own cost. And so that he can maximise the cost from there.

Now, so regarding these things, this problem can be expressed in the optimization problem statement which is here. Basically, the idea is that. This is the objective function maximize, maximize is basically here. c_i if the i-th item is selected and w_i is the cost and x_i is basically how many of that element is selected or it is selected yes or no. x_i is basically if it is selected then 1 if it is not selected then 0 and c_i and this one. This is basically the objective function in this regard. And then constant is that constant is. Basically, x_i into w_i should be $\leq w$. That is the constant, because if you select one item x_i then weight of these things and summation of all the selected items should be within the maximum limit of the limit of the Knapsack that is the w . These are the problem and the statement of the optimization problem and having this optimization problem, let see how we can solve it.

(Refer Slide Time: 16:54)



First, we will see that how it can be solved using naive approach and then finally, how the same thing can be solved using GA approach or basically we will decide about encoding scheme. Now, here this pictorially once simple instance where the different items for example, item 1, item 2, item 3, three items only are there for this item 1 the weight is 10, 20 and 30 units and the cost of the item 1 is 60-dollar, 100 dollar and 220 dollar and this is the this this is the Knapsack this is the Knapsack the capacity of the Knapsack is 50. We have to take some items selectively from there. So that we can have the maximum this one.

Now, as an example if we select 10 and 20 then; obviously within this capacity, but the solution gives the cost is 100 and 60. Or say, 20 and 30 it is possible and then it will give the cost is 100 and 120. This is like this the different solutions.

Now, what are the different solutions are possible it is listed here. We can select single item at a time and then two item at a time and then three item at a time. However, this is not a feasible solution. Because, if we can include the item 1, item 2 and item 3 , the cost may be 10, 20, 30 but it will exceeds the maximum quota. Because, the maximum weight is 50. It is not the feasible solution.

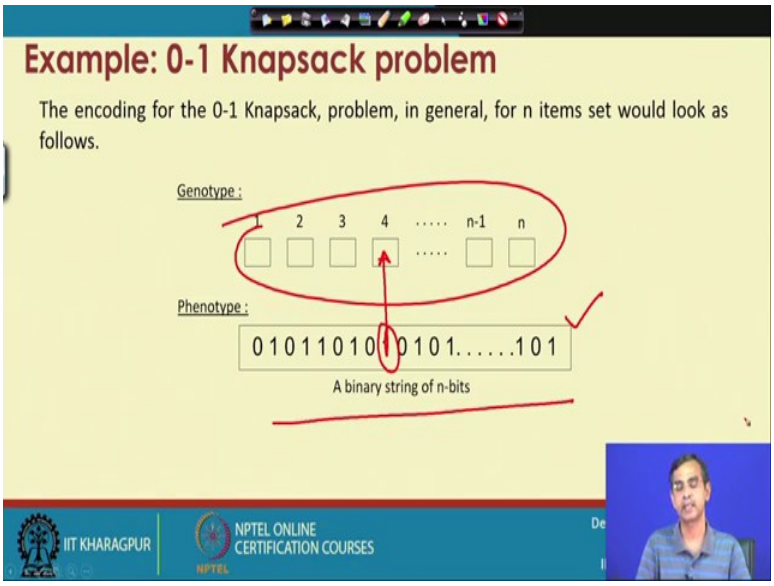
Now, what I can understand is that is basically the problem, if we solve in a naive approach. Out of the n items we have to select all possible subsets of this n items. The number of possible subsets that is there is basically 2^{n-1} . That means, for this problem there are 2^{n-1} solutions are possible. N if the n items are given to you. Out of this 2^{n-1} there maybe 1 or more solutions are the optimum solution; that means, it will give the maximum the objective values.

Now, here as a genetic algorithm task is to search for this optimum value. Now, how we can decide the encoding scheme? We can decide the encoding scheme like this if one item is selected then we can represent. There are n items. We can decide the n items we can decide the n items and then the length of the chromosome is n . It is basically item 1 and item 2, item 3 and so on, on n item.

Now, if an item is selected then we can press 1 if it is not selected 0 it is like this. This way the n out of n items the subset of n items which have been selected for the solution is a one chromosome. This way you can represent the chromosome.

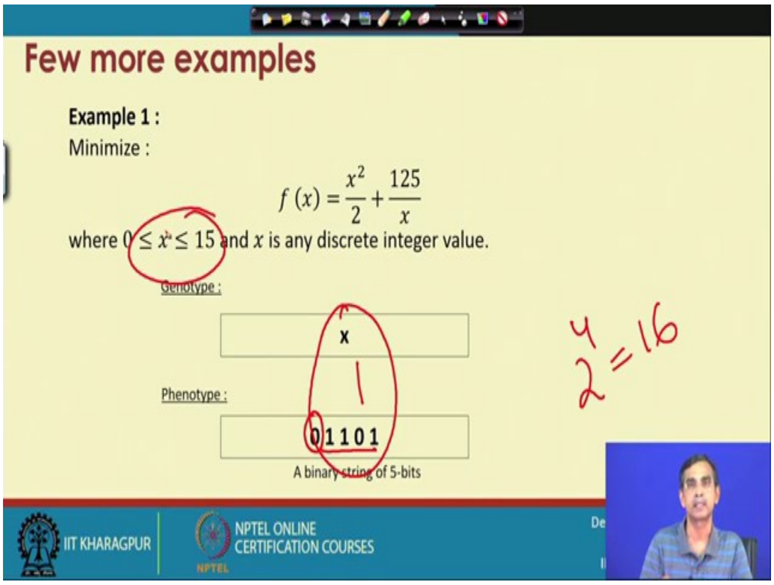
Now, for example, in this case the number of item is 3. The chromosome length is three and the different chromosome that is possible in this context is shown here. As the n is 3, the different number of chromosome in this case is basically yeah basically 2^n all possible permutations you can see. This one are the different, different chromosomes representing the different solution. What we can say this is the one solution. The solution is basically same as this one this another solution represents this one and similarly this is the another solution is this one. Different solutions can be encoded and this is basically the encoding scheme in this case this is a Knapsack problem.

(Refer Slide Time: 20:56)



Now, the encoding scheme for the 0- 1 knapsack problem in general for n items has the length of the chromosome. Chromosome is n, it is a binary string of n bits and this is the structure. This is basically a particular instance; that means, a particular individual we can say or a solution and the solution is called phenotype and this is the genotype this means that any i-th bit is basically represent the i-th parameters. Here n parameters basically to solve the problem whether one particular items is to be selected or not is the concern.

(Refer Slide Time: 21:42)

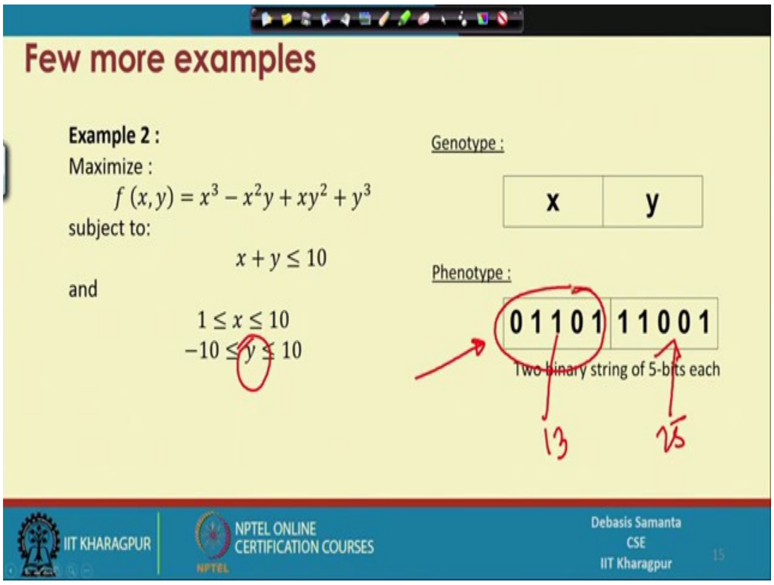


Now, let us practice this concept with few more example, I want to give one simple example here. Suppose, this is the one optimization problem where objective function is defined here and this objective function is to minimise $\frac{x^2}{2}+\frac{125}{x}$. That mean we have to solves the values of x for which this function $f(x)$ has the minimum value. And suppose the range of the values is in this range x ≥ 0 and ≤ 15. And suppose any discrete integer value that mean 0, 1, 2, 3 are the values not the real values.

Now, if this is a problem as the it contents only one parameter namely x. It is genotype consist of only x. This is a genotype only very simple and so far, the phenotype is concerned we have represented one phenotype it is like this. It has the value say 1 and then 4 it 13. This is basically representing 13 the equivalent value in decimal.

We have understood it. Now you can see one thing that you can note it. For x within this range 0 to 15. There are different values is basically 16 and for the 16 any discrete values integer values representation we need maximum 4 binary bits I mean 2 to the power 4 this equals to 16. In order to represent 16 numbers uniquely, we need the 4 bits or binary string of length 4. Here, we have considered; however, 5 absolutes no problem in that case only this is the MSB most significant bit will be 0. Within this 4 bit we can represent. Minimum requirement is 4. And we have considered here 5 we can consider of course, but minimum if it is possible then we should go for this; that means, this x can be represented with 4 bit. For this constant is concerned.

(Refer Slide Time: 23:58)



Now, as another example this is another example little bit complex compared to the previous one. And here if this is the objective functions to be minimised and you can note that this objective function is in terms of two parameters x and y. Your design parameters is two namely x and y and this is the type of the objective function we have to write. That means, we have to find the good values of x and y for which this f can gives us the minimum value.

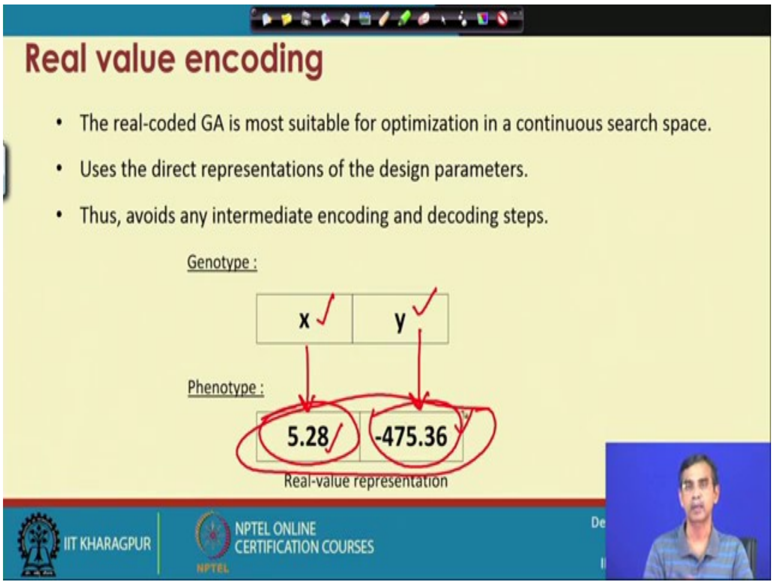
Now, here again there is a constant, the value of x and y should be chosen in such a way that it will say it will satisfy this inequality. That mean $x+y$ should be ≤ 10 . Now, without any special thinking and here also we see other different what is called the range of values that x and y should be $1 \leq x \leq 10$. 10 different values and $-10 \leq y \leq 10$. So, 21 different values.

Now, here 10 different values we can easily use I mean for the representation of x the 4 binary bits; however, I have used the 5 binary bits absolute no problem. Similarly, for the 21 we can represent the 21 different numbers within the range minus 10 to 10 by another 5 binary bits it is the 5 represents.

Now, at any instance it basically at any instance this is basically one instant or one individual or one solution. At any instant, the value of this binary bits here is basically 1 and then 4 and 8, 13 this basically 13. And similarly, this basically 16, 20 this is basically 25, 25 it is 1, 2 this is not correct anyway. This is this represent some 25 may be. This represent 20. However, this is not in this range when you will check it this will be excluded. Anyway, these are the different genotype and then concern is the phenotype is there we have discussed it.

Now, we have discussed about the different encoding there the binary encoding scheme.

(Refer Slide Time: 26:20)

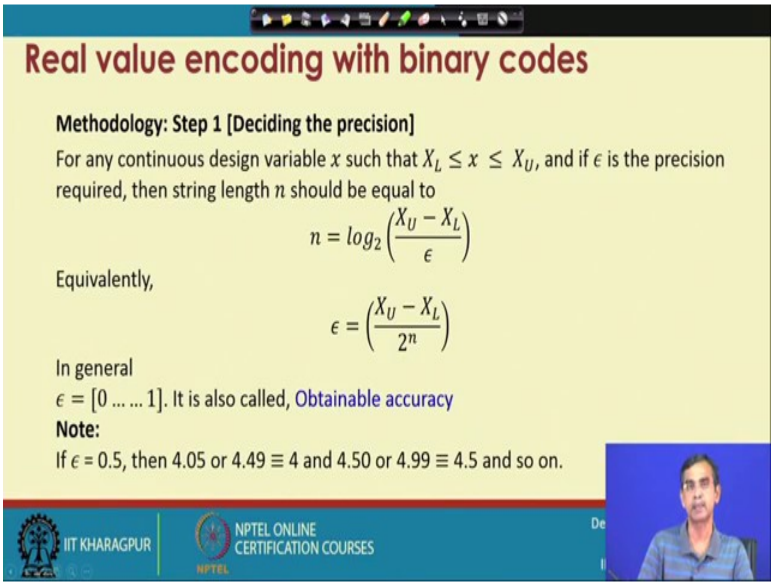


And now, we will come to the discussion of real value encoding scheme. The real coded GA is more suitable for optimization in a continuous search space. And it basically uses the direct representation of the design parameter unlike in case of binary encoding scheme, there is a need to convert any value into their binary equivalent. But, here no need to represent in a binary. It is a straightforward.

Now, for an example if an objective function is consisting with two design parameters namely x and y. And they are values at any instant say x is 5.28 and y is this one. These constitute the one what is in combination or phenotype or is a chromosome. It consists of two values right one is 5.28 and this one. This constitute a solution; that means, at any instant the solution is that $x=5.28$ and $y=-475.36$.

Now, the real value encoding scheme is very simple.

(Refer Slide Time: 27:28)



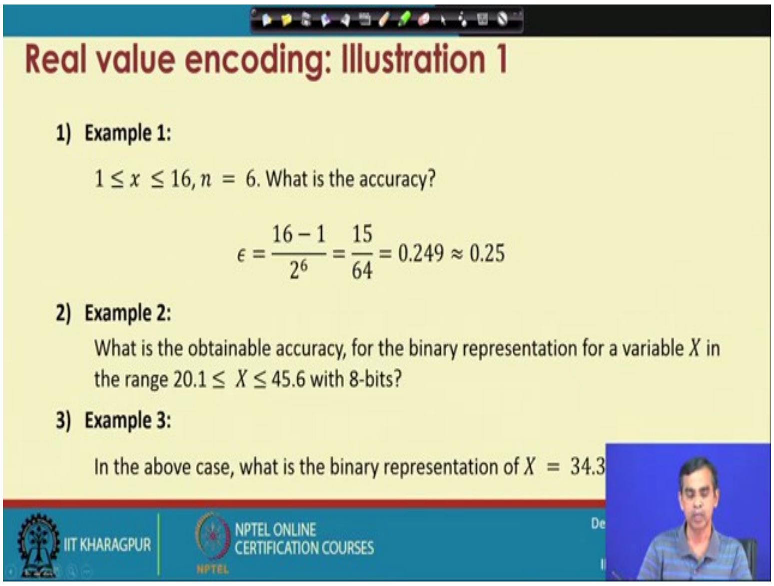
However, real value encoding also can be in the form of a binary course it is the more usual practice basically. This is because, the binary encoding scheme is faster and gives more accurate results. That is the many users many programmers they prefer the binary encoding scheme. And in 2-3 slides I will quickly give an idea about how the binary encoding can be adapted in the real value encoding scheme. It is basically use some formula.

The formula is that first you decide to represents a real value. How many binary bits is required at least? So, this basically gives a formula. This formula gives that how many binary bits is required to represent a value. And if the value has it is range from X_L to X_U. Where X_L is the lower range and X_U is the upper; that means, if a value is lying within the value range X_L to X_U then n can be decided by this one. And here one important factor epsilon, epsilon decides there what is the obtainable accuracy? I mean how much accuracy that you want to have it.

Now, from this expression $\in$ also can be calculated within this formula now. Within this thing obtainable accuracy can be calculated and then based on this obtainable accuracy, we can decide n the number of bits that is required to represent a real value. For example, if $\in = ¿$ 0.5 then 4.05 and 4.49 all the values will be within this range will be represented by 4 it is this one.

On the other hand, if it is $\in = ¿$ 1 then 4.00 to 4.99 should be represented by 4 like this one. Depending on the obtainable accuracy the range or precision will change and accordingly the number of bits will be decided. Now, here this is the formula that we should follow in order to understand how many bits is required and with a desirable accuracy it is there.

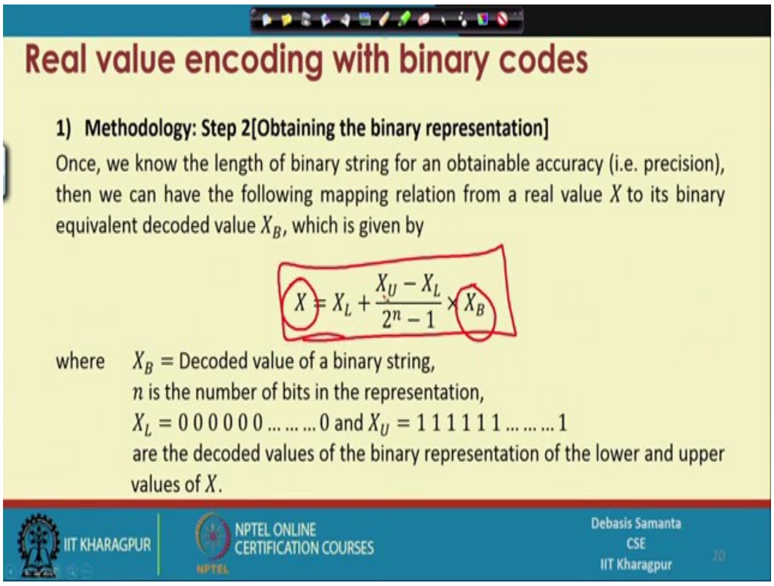
(Refer Slide Time: 29:34)



Now, this can be followed to solve it as an example, we say suppose x is within this range and n is the 16; that means, we decide the number of bits that is 6 then obtainable accuracy if x is within this range then it can be call 0.25. Within the range of the values and then n we can decide the obtainable accuracy and finally, we can use this one to calculate the number of bits. For example, you can easily calculate, with 8 bits and if the number is within this range and what will be the optimal accuracy?

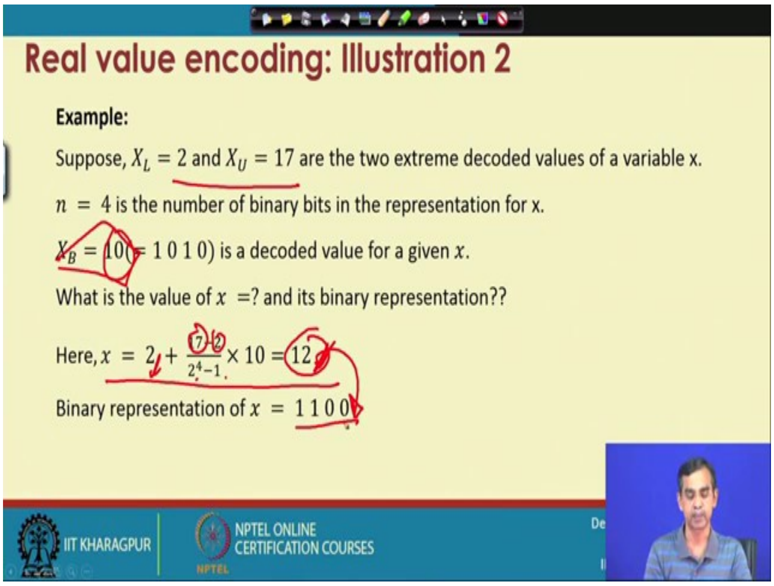
Now, for example, say suppose $x = 34.35$ is a representation then what is it is corresponding binary scheme, binary value? Now, let us see how this can be obtained it can be obtained easily.

(Refer Slide Time: 30:27)



Now, this is the formula that can be used to understand that if X_B is the current value and then X is basically it is a binary representation. How it will be this is the standard formula that is followed here and n is the number of bits in this case.

(Refer Slide Time: 30:45)



Now, for an example we can understand about say $X_L = 2 \wedge X_U = 17$; that means, lower limit and upper limit of the parameter x . And n is the 4 that means, number of bits that is required and say suppose X_B the binary value that is 10. Which is basically in the binary representation can be like this 1 0 1 0 0 2 then 4, 8. It is 10. This is basically the decimal equivalent, this 10 is the decimal equivalent of this binary values.

Now, having this one what will be the x that can be represented if it represents X_B . This can be obtained using this formula. Here, this is basically the X_L and this is

$X_U - X_L$
¿
¿
¿ into this is the current value X_B . It is 12. That means, this this X_B which is 10 actually it is 12. For the real value is concerned in the range 2^{17} and this 12 can be represent using binary it is called 1 1 0 0 representation.

Here, is the idea about that within a particular range X_L and X_U and given the obtainable accuracy will be able to represent any binary value in the binary encoding scheme. With this things, we have learned about the binary encoding scheme and in the next we learn about the other encoding scheme binary encoding scheme and the real value encoding scheme is covered, the order encoding scheme will cover in the next lecture.

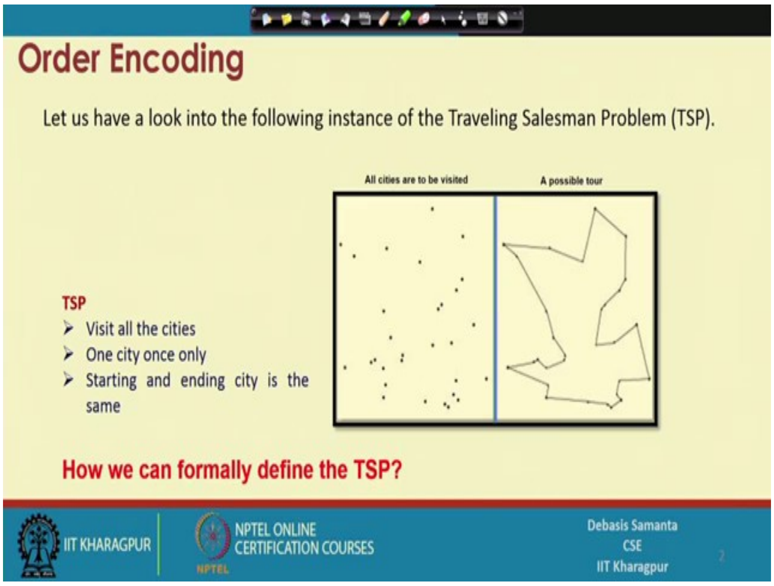
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 17
GA Operator: Encoding schemes (Contd.)

In the last lecture, we have learned about two encoding scheme the binary encoding and real value encoding. Today, we are going to learn other two encoding scheme namely order encoding and tree encoding.

(Refer Slide Time: 00:36)



So, the order encoding can be better understood with an example. So, I want to give one example this example we cited from the famous problem it is called the travelling salesman problem.

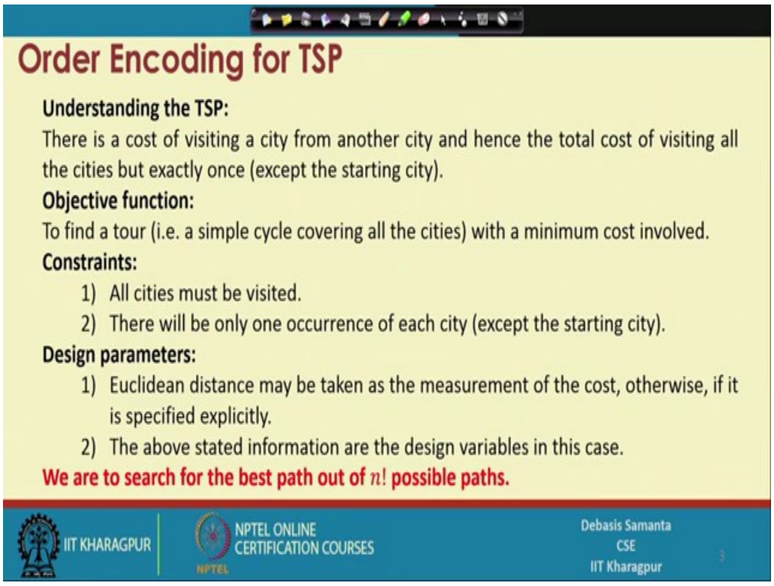
So, in the travelling salesman problem, the problem is defined like this, there are n number of cities and some cities are connected to other some cities. So, that one travellers he can travel from one city to another and so for the travelling from a city to another city there is a cost involved. So, the problem is to find a path or a route that the traveller should follow. So, that he will incur minimum cost of travelling, but the constant is that he should travel all the cities, but exactly once and he should return to the starting city at the end.

So, this is the problem and this problem is called travelling salesman problem and famously it is called the TSP and in the TSP, this figure shows basically a simple way of representing the different cities. So, that location of the different cities on the surface of the earth and one path that it is one tour rather we can say one tour for a travellers is shown here, say is this is a tour.

So, if there n cities are there, then essentially we have 2^n different tours are possible. That means, there are 2^n number of solutions are possible and out of these 2^n number solutions we have to search for the optimum tour. Optimum tour in the sense that the tour which required minimum cost for the traveller.

Now, this is a problem and optimization problem also little bit understood. So, now, let us see how this optimization problem can be defined and then corresponding it is encoding.

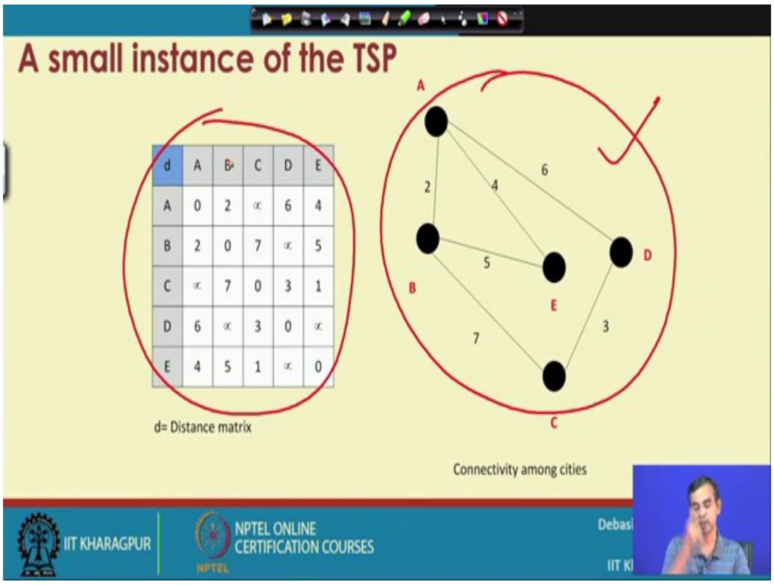
(Refer Slide Time: 02:44)



So, here is the idea about how the optimization problem can be express, that is fine. So far the optimization problem is concerned. The input is we have given the different cities and the cost of travelling from one city to another. So, this is the input that is given and also it is to mention that which is the starting city. Then, objective function is basically to find a tour or rather we can say a cycle covering all the cities exactly once, except the first cities and with the minimum cost involved and the constraint in this problem are all cities must be visited and there will be only one occurrence of each city, that means, he should not visit the same city more than once except the starting city.

And here, design parameter we can consider about if the location of the cities is given and then Euclidean distance between the two cities can be taken as the cost. Otherwise, it is spaces if it is specified explicitly there what is the cost of travelling from one city to another and then this problem can be stated in terms of mathematical representation, it is like this.

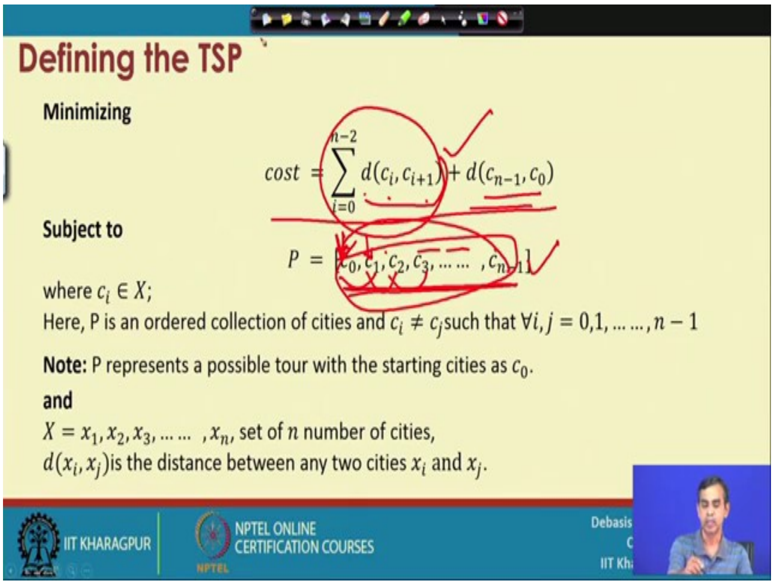
(Refer Slide Time: 03:59)



So, this is basically the input, this input say here this problem is simplified for five cities, the cities are termed as A, B, C, D, E and then is the cost of travelling from city C to D it is basically is a unit is 3 likewise and if there is no path from one city to another, then the cost of travelling is a very large, so it is infinite. For example, so from city A to city C there is no path. So, here this is the same representation which is shown in the form of a matrix it is represented here, from city A to city C we do not find any path, that means, we can say that this is the cost of is huge or infinite cost like this.

So, the idea is that, if this is essentially the pictorial description of the city map and the same concept it is stored here in the form of a matrix. So, this basically the input to the problem. It is the cost matrix like this one. Now, having this is the problem statement, we can define one encoding scheme like, so order, that means, what are the different possible ordering that it can have.

(Refer Slide Time: 05:21)

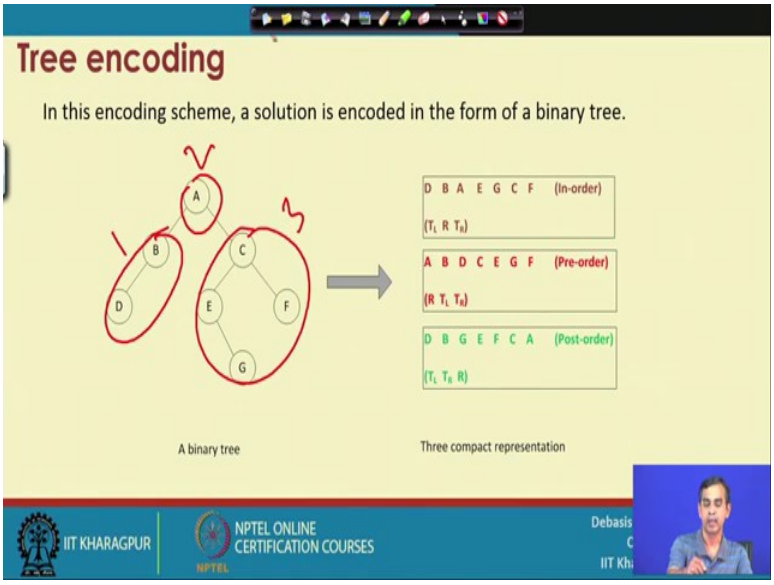


Now, regarding this ordering as a simple example, fine first before going to this ordering scheme again let us first little bit explain about the objective function. Here, the objective function is defined using this formula; here d the distance from any city c_i to c_{i+1} and if we at any instant, so this is basically the at any instant, this is the solution, where the solution is that that order first start with c_0, then c_1, then the city in sequence.

So, this is basically ordering of the different cities are there and then if this is the solution then we can evaluate the cost of this even this formula. So, c_0 to c_1 what is the distance, then c_1 to c_2, then c_2 to c_3 these are the so sum of all distance and then the finally, c_{n-1} to c_0 because he has to return to the starting city then. So, this is the distance of covering from $n-1$ to this one.

Now, so if this is the one solution and then we can obtain the cost of the solution. So, here encoding scheme, this is the encoding scheme that we have followed, it basically the sequence of visit, visiting the cities. So, sequence means if city A is first, then city D, then city B, then city C. So, it is A, D, C, B like this, so it is the different sequence of there.

(Refer Slide Time: 06:45)



Now, so this way we can encoded and this is the major the main concept in the order sequencing. So, this is pretty simple actually, so what are the different sequence it is possible and then that is the order encoding scheme. Now, after this order encoding scheme, we will discuss about another encoding scheme it is called the tree encoding scheme.

Now, the concept of tree encoding scheme it follows the concept of tree in this slides I have shown one tree, here A is basically starting node and from A there will be two what is called the child the B and C, similar C being the node it has two children E and F, for the E only one children it is called the right children and there are no left children, for B similarly it has left children and no right children. So, it is a typical form of tree and here we can note that one parent here has at most two children. So, it is in some sense it is called the binary tree.

Now, having this is the binary tree, it has basically representation of certain what is called the way of I mean storing the data, if all the values in this node represent some symbols or data for example, A, B, C, D in this case, then in the order they will be visited it basically called one sequence or one encoding.

Now, I have mentioned three different ways the tree can be visited and the first method is called in order. So, first we will visit, so in order means to visit this things, we have to visit these things first, then we will visit this one and then we will visit this one, again the same way to visit this one we have to visit this one, then visit this one, then this one likewise.

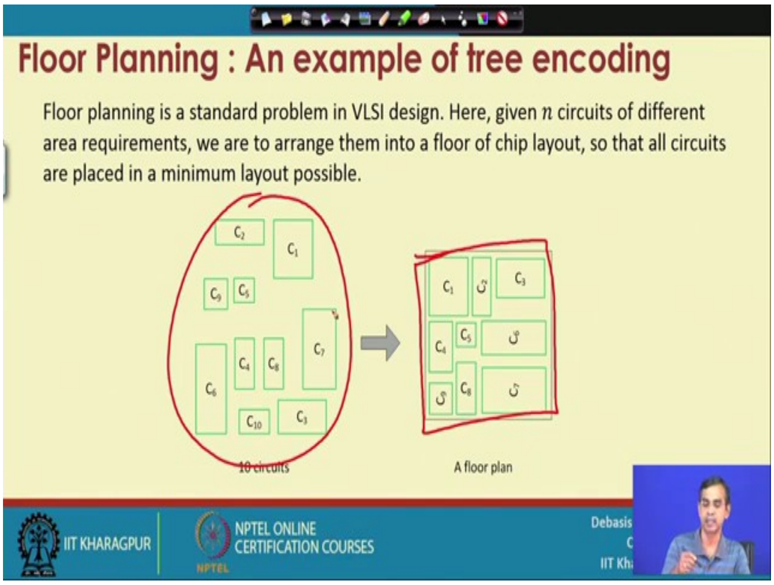
So, if you follow the same procedure, same policy for each then the visiting will be there and then it will be listed. For example, so if we consider the visit that we will visit these things first, then this thing and then these things. So, visit this one, then this one and then this one, it is called the one sort of visiting and this is one sort of encoding in fact. Now, for using this one, so the idea it is that we will visit in order to visit the tree we have to visit this one, to visit this tree we have to visit this one. So, first D visited, then we have to visit this one it is empty. So, no nothing to mention here, so the next B, so this completes the visit.

So, these visit is basically D and B, so it is basically the left tree is visited left part visited, then we visit this one. So, it is A, then the right part will be visited again visiting right part we have to visit this one, now visiting this part means, so it is empty. So, no need to go there, so we will visit E then G, so basically E and G. So, visiting this one, then we will visit C, so it is C and then finally, the F will be visited. So, this is the order, so it is called the inorder traversers.

Now, similarly preorder traversal means we will visit this one first, then this one and then this one. If you follow it, we visit this one first, then visiting this one mean B first, then D, so BD. Similarly, visiting this one means C first C, then visiting this one means E first E and G and then finally F, so this way the different thing.

So, what I want to say is that if this is the tree that is given to you, then these are the different way the tree can be visited and each way representing one, what is called the encoding form. So, this way it is the tree encoding come into the picture and it is also some way resembles with the order encoding scheme that we have discussed for the genetic algorithm.

(Refer Slide Time: 10:47)

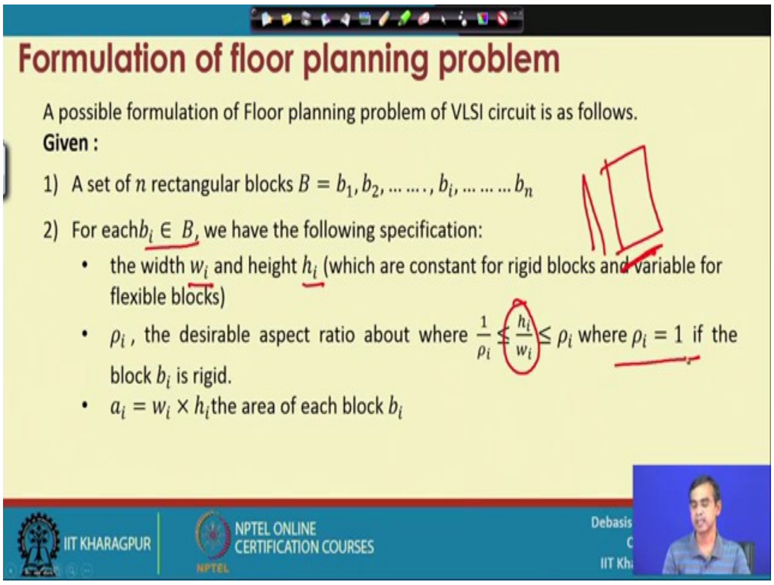


Now, so this is the tree encoding scheme and this is one encoding scheme is there and next I want to give one example of this tree encoding scheme, this example is basically a is a problem it is called the floor planning problem. So, floor planning problem is basically the idea is that the different what is called the blocks are there.

So, these are the input blocks of different sizes and you have a floor and then you have to arrange all these blocks into this floor. So, that the there will be no wastage of space and within the minimum floor area we can place all these blocks into the floor. So, it is basically layout of the floors needs to be decided by placing all the blocks into that floor. So, that it takes the minimum area or something else like.

Now, this is the one example called the floor planning and we will see exactly how the tree encoding can be applied to solve this kind of problem.

(Refer Slide Time: 11:54)

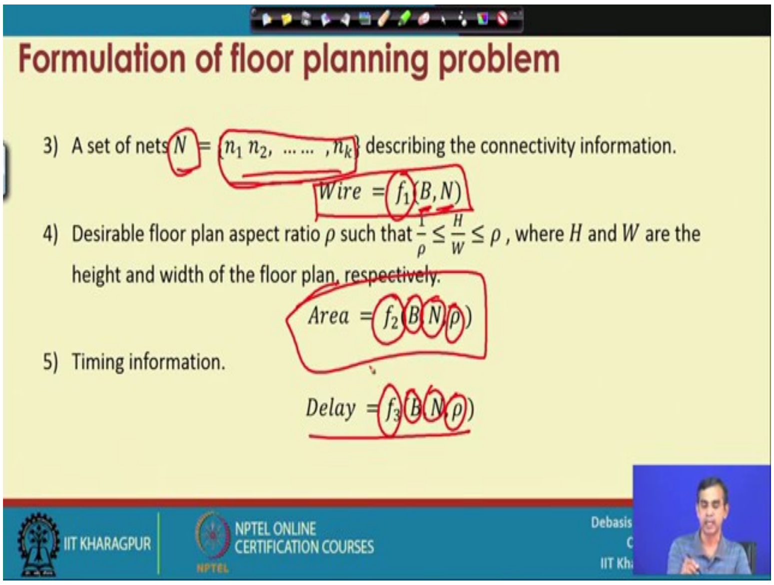


Now here, to be more specific for the problem is concerned we want to make a little bit complex. So, that the problem has taken it is own strength like.

So here, the input is a set of blocks. n number of blocks and we assume that all blocks are of rectangular size and these blocks are denoted as $b_1, b_2, upto\, b_n$. So, these are the n number of blocks are the input to this problem and for each block there is some specification. The specification is there. For each block it is specified by width and height. That mean, if this is a block it basically width w and height w, so it is specified and there is also another specification. So, and then blocks are rigid, rigid means no a, this width and height cannot be changed and ρ_i for each block it basically states the aspect ratio; that means, it is basically decided ρ_i this one, $\frac{h_i}{w_i}$.

So, if $\rho_i = 1$, if the blocks b_i is rigid, otherwise it is n. So, we assume that ρ_i ¿1 this case and then given the w_i and h_i we can easily understand our area of the block.

(Refer Slide Time: 13:19)



So, a_i denotes the area of the i-th block or block b_i. So, this is the specification of the problem at n and then we will see exactly how the problem or what is the objective function for this problem is there.

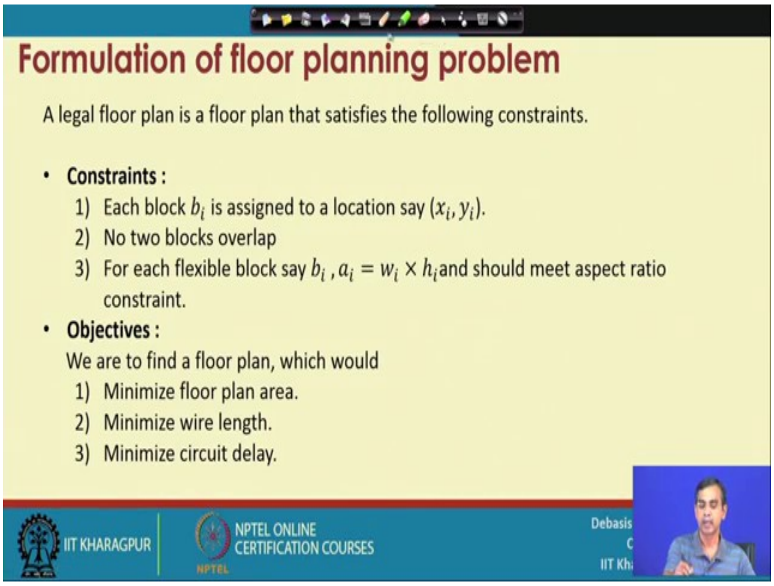
Now here, so there are n number of blocks and we assume that the blocks are connected from there is a connection or you can say line from one, what is this is basically the problem related to the VLSI problem, it is called the very last scale integration where we have to place the different circuits. A circuits resembles to one block and then there is an input output connection. So, this connection is basically called the net, so there are N number of connections which is shown here, the set of connections; N is a set of connection, these are the connections.

So, the idea is that if B is given there and N is the connection and f_1 is one objective function it is known there, by which it can say what is the total wire that is required, wire means how many connectivity are there, how many cost of networking is required and also if rho the aspect ratio is given for each block and then set of block is given and then this is a network connection then what is the area that is required it is also given by this function f_2.

Similarly, f_3 is another function, it will basically calculate delay of the entire things for so far the circuit is concerned. So, it is given the set of block and then connection and then rho the aspect ratio then, then f_3 this will give the measurement about the delay involved in the layout design. So, these are the three functions f_1, f_2, f_3 it is obtained from the VLSI specialist, they given this B and N; B and N, ρ or B, N, ρ, they will be able to calculate f_1, $f_2, \wedge f_3$, namely the wire length, the area and the delay of the block that is required.

Now, so our objective, so far the genetic algorithm is concerned find a network connection out of the so many networks and from the B, so that this f_1 can be minimised, f_2 can be minimised and f_3 can be minimised. So it is basically the multiple objectives here, unlike the single objective that we have discussed so far.

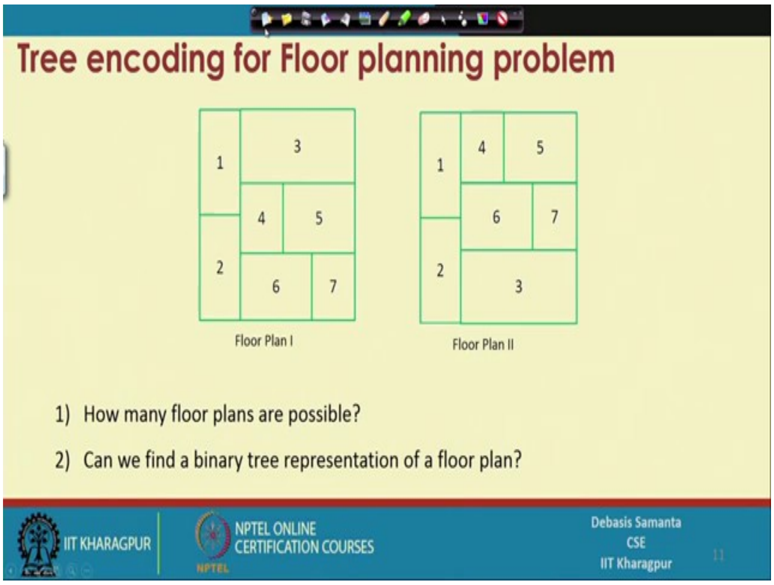
(Refer Slide Time: 15:34)



Now, in this multiple objective, we have to we have to minimise three objectives, the area and length and then circuit delay. So, these are the thing defined by the function it is already discussed.

Now, we will come to the encoding scheme rather. So, how the encoding scheme can be there?

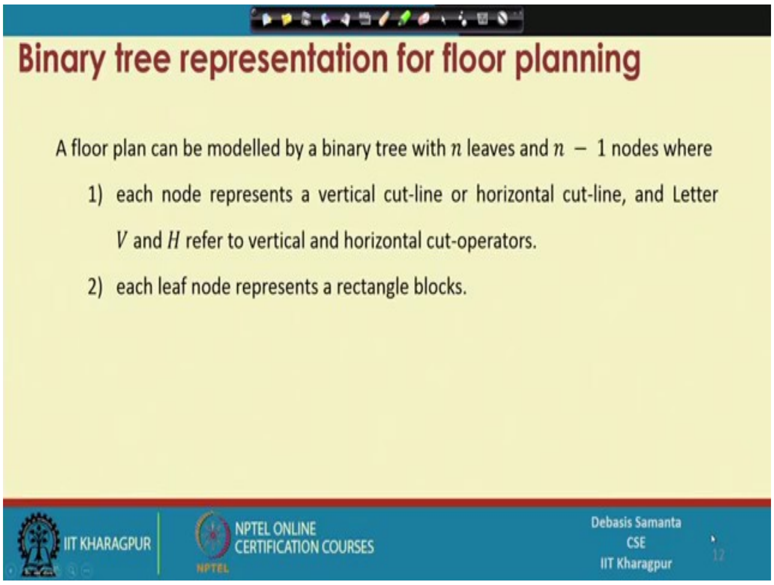
(Refer Slide Time: 15:53)



Now, before going to the encoding scheme let us consider two instances here. So, these are the different block 1, 2, 3, 4, 5 like this on and these are the one plan, so that if it is plan like this. So, for the same set of blocks two plan we have mentioned it was and this is the 1 plan. So, for this is the two floor plan are given here and then we have to see whether for these floor plan area is minimum or delay is minimum or the length is minimum whatever it is there. So, basically the idea it is there.

Now, given a set of block can you imagine how many different floor plans are possible? There in fact, many, that is a, and searching for all the floor plan and then ultimately finding the minimum is a very time containing problem. So, this can be represented in GA frame work and then finally, can be solved using genetic algorithm.

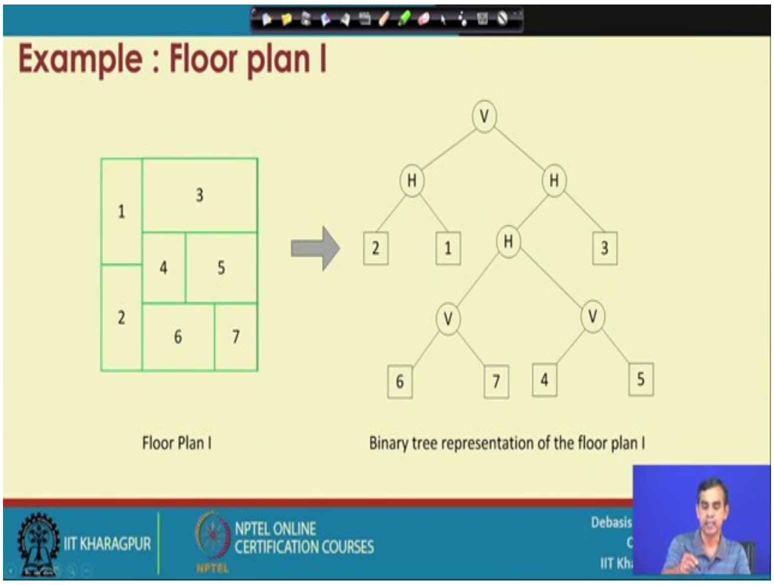
(Refer Slide Time: 16:54)



Now, so to do this thing, we have to do certain encoding scheme. Now, so encoding scheme that we are going to propose it basically like this. It can be modelled in the form of a binary tree because we are going to realise it is in the tree encoding with n number of what is called the leaf nodes; that means, the last node does not have any children and $n-1$ the non-leaf nodes and each node represents a vertical cut line or it is called the horizontal cut line and later they can be represented by V and H respectively and each leaf node represents a rectangular blocks, so this is the specification.

Now, let see how the same thing can be represented in the form of a tree given floor plan like. So, idea it is like this.

(Refer Slide Time: 17:40)



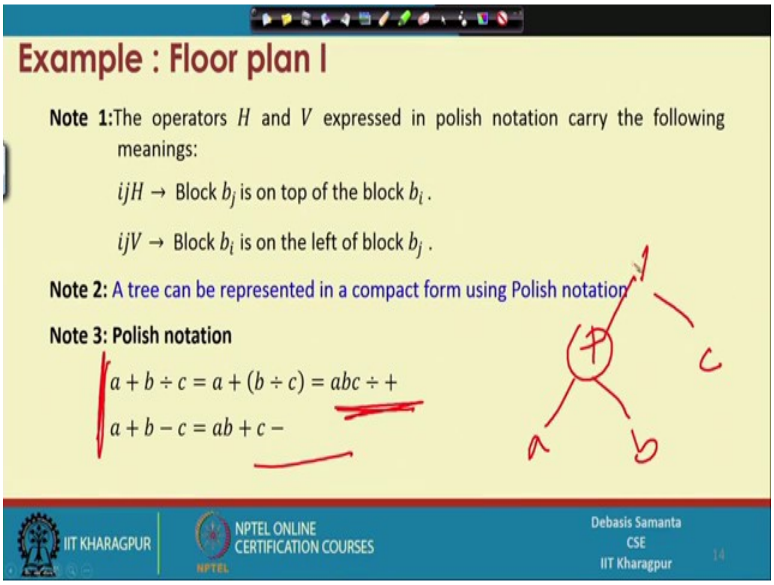
Now, here you can see it. So, this is the one floor plan and this floor plan can be represented equivalently in the form of a tree, so here V basically vertical line. So, it is basically the vertical line and then there is a one horizontal line, so it is V then H and then 2 and 1, so these are leaf node.

Similarly, if we come to this part, so this is basically left part of the tree. Now, these are right part again from one vertical, so right part, this is the right part, so these are right part. Now, in this right part we first have the horizontal cut. So, this is the one horizontal, so this H and these 3, so this is the 1 and then again this part, so this part again another tree, so it is represented by this part. So, here one horizontal this is the horizontal cut and this horizontal has one vertical. So, it is vertical 4, 5 and another vertical 6, 7, so this one.

So, idea is that, this kind of floor plan can be represented by means of a tree whose structure is here. Here, these are the leaf node, Leaf node represents the block ultimately and other the non-leaf nodes V, H , they are basically whether the vertical cut or horizontal cut. So, it is a vertical cut, it is a horizontal cut back on so.

Now, so the idea is that any such floor plan if it is given you, we can find is an equivalently the corresponding tree and this is the concept of tree encoding in this case, now as a few example further.

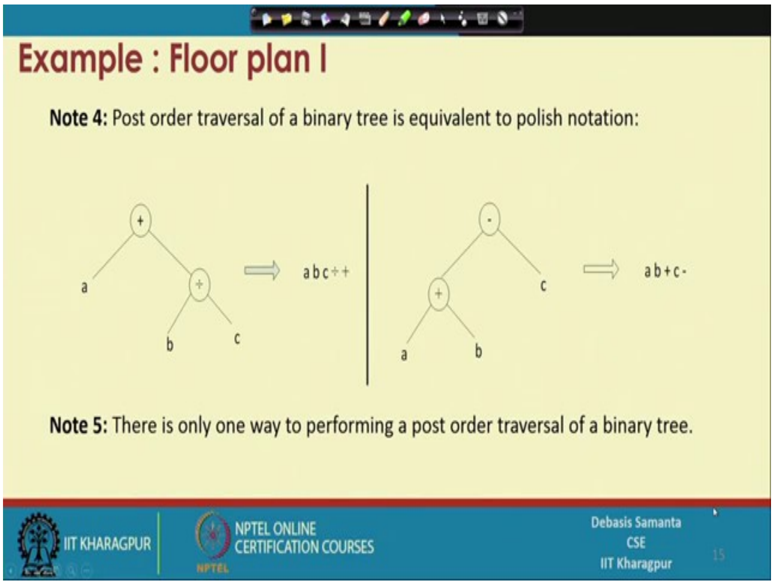
(Refer Slide Time: 19:22)



The different floor plan if it is given to you, you will be able to represent it and the tree can be represented using the different traversal that we have discussed inorder, postorder and preorder. So, specifically one particular order is used that is called the postorder traversal; postorder traversal can be expressed in this form.

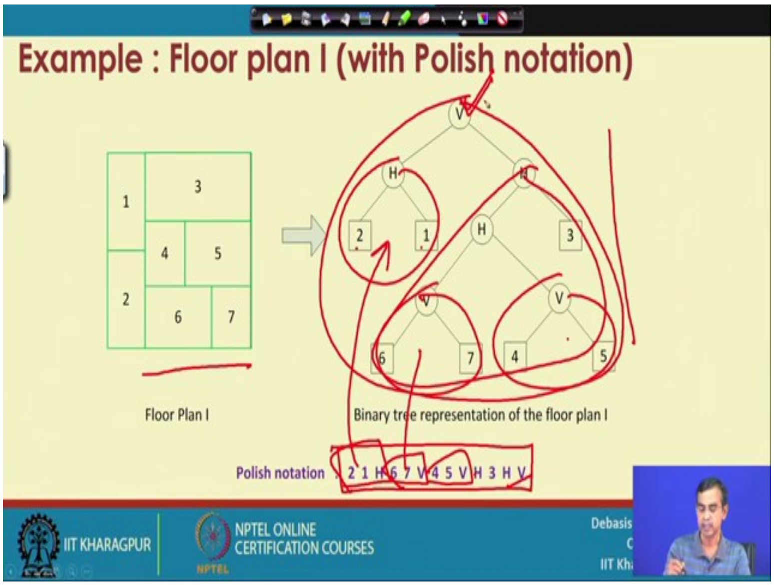
So, if it is like $+a, b$ then /, then c, so is $a+b/c$ this kind of thing. So, it is basically $a, b+c$ this one, so it is like that. So, this is the way that can be, so anyway some tree representation can be used and then the one traversal the postorder traversal can be followed and then it can be obtained.

(Refer Slide Time: 20:14)



We can represents one example it is like this.

(Refer Slide Time: 20:18)



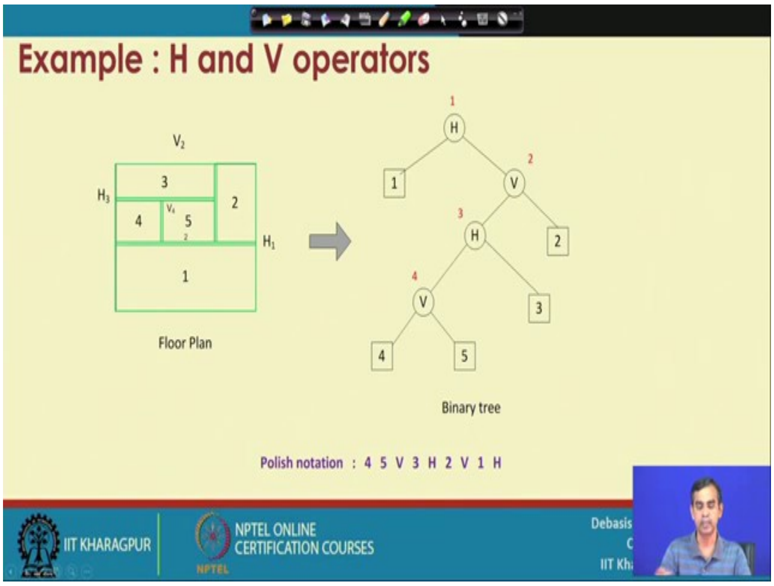
Here for example, here say, suppose this is the floor plan and this is the equivalent tree. So, this tree is a graphical display, this can be also alternatively represent in this form, it is basically called the polish notation or postorder traversal; postorder traversal means, so in order to transverse it first left, then right and then finally this one.

So, here for example,. So, first left means we have to find this 1, now to do this again left, right this one, so 2, 1, H so far the this traversal is concerned, then we have to traverse it. So, to traverse it first left and this again left, so 6, 7, V, so 6, 7, V corresponds this traversal first, then this one it is 4, 5, V and then finally, it is H this 1 and then finally, will visit 3 and H. So, it is this one and finally, the last one. So, this way, so this basically if this is the tree given to us, then it has the equivalent representation which is shown here.

So, what I want to say that, if this is the floor plan and if this is the tree encoding, then this tree encoding can be come back to a represented by this one. So what, in other way, other what is it is that, this is basically is an encoding scheme for a solution. So, this is the solution and the solution is represented in this form and this is the encoding scheme and this in fact is a solution means it is the chromosome.

So, in this case the length of the chromosomes same as the number of nodes involved here. Now, the number of nodes can vary if the different order is followed. So obviously, it is the variable length chromosome can be obtained if it is like this.

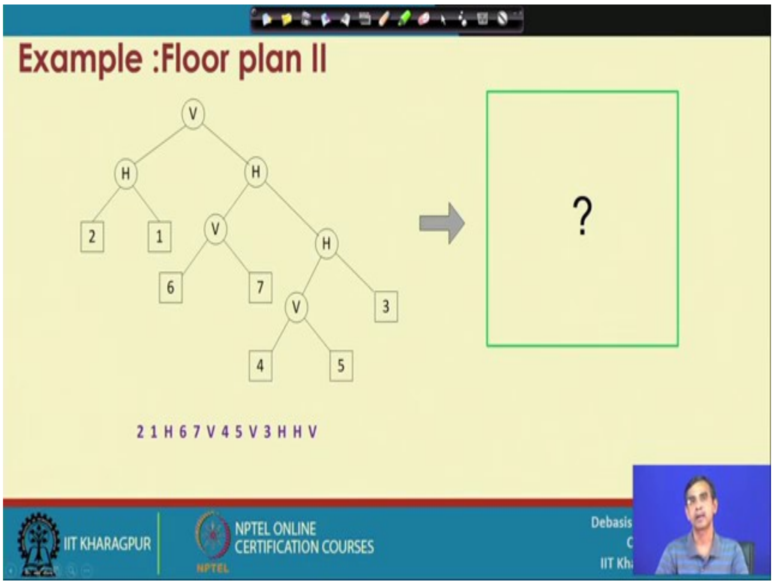
(Refer Slide Time: 22:02)



Now, so the same idea can be extended and then you can for example, if this is the one layout it is given then corresponding for your practice you can check it, if this is the like one solution this is the tree and this is the solution representation or encoded form of the solution or a chromosome, so the idea it is there.

Now, the problem here, our objective is to solve the optimization problem.

(Refer Slide Time: 22:30)

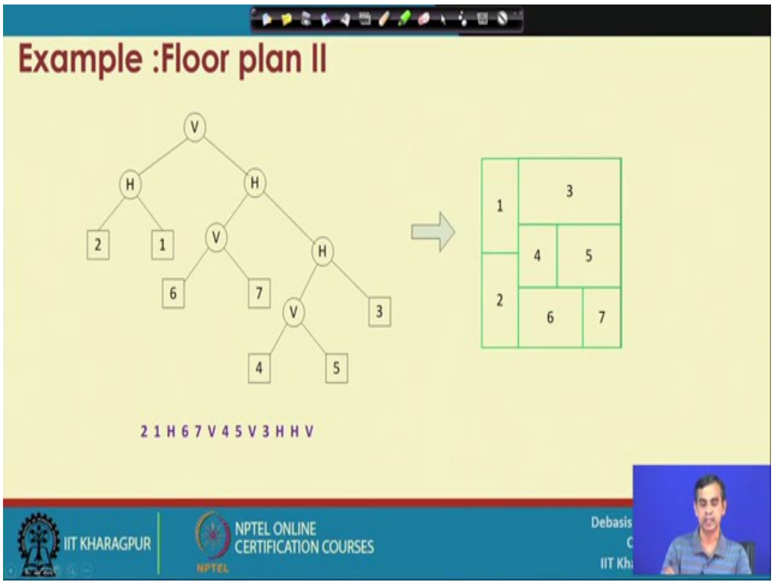


So, that how we have to find the best solution here. Now if, so alternative also if this is given we can draw the floor plan also; that means, if it is given we can do it or if it is given we can obtain it and then we can obtain the floor plan or directly from the floor plan. So, all the representation is basically same and then same thing can be obtaining uniquely.

Now, so idea here is that, this is the one floor plan we have to obtain, then one solution is tree that means one representation, that can be thought of; that means, how many way this representation can be thought of, of if the n number of nodes are there, so basically the number of solutions are, basically how many different tree that can be formed.

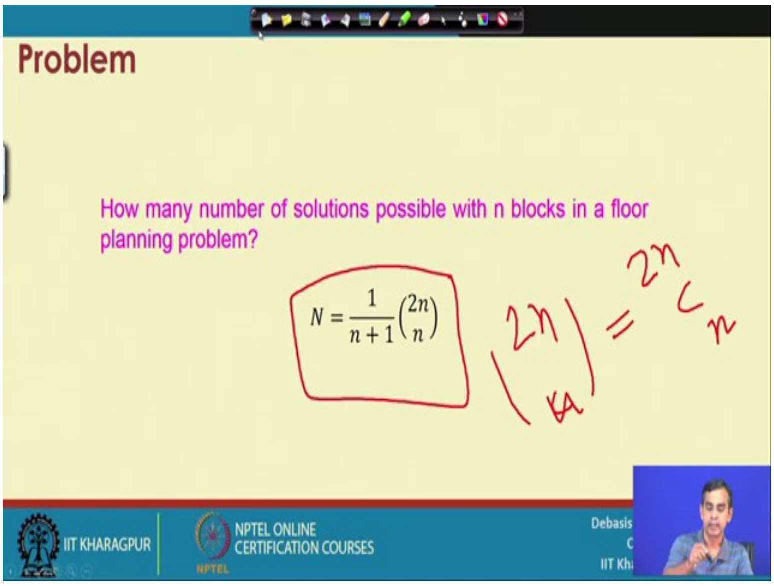
Now, so if it is only fewer number, then no problem, but actually this number is a large number. That means, for n number of blocks the number of trees that is possible which represents one solution is a very large number.

(Refer Slide Time: 23:32)



So, this number is, so this is the, another example you can understand whether this is correct or not, this is the tree representations, this is the floor plan and this is this one.

(Refer Slide Time: 23:39)



Now, the question is that how many number of solutions possible if there an n number of blocks in a floor. So, if floor plan with n number of blocks, then how many solutions are there? Now, it is very difficult to calculate the number of trees that is possible and it can be shown that the number that is possible is this one. So, so this, this is even eventually a very large number for N where $2n,n$ basically it is it represents $\binom{2n}{n}$, this is equals to basically combination $2n_{C_n}$.

So, this is very large number in fact, and then, so it is a large number means we cannot solve in real time. So, it requires some other approach or hard programing approach cannot try out all possible trees and then finally, finding the optimum tree; optimum tree corresponding to the, the best floor plan in this case.

So, the genetic algorithm approach is, we have to randomly choose some solutions, randomly choose solutions means, whatever the pattern that we have discussed about there one pattern or one path or tour and then decide the solution and this way we can try out all possible in a random or probabilistic search manner.

So, this is the concept of tree encoding that we have discussed about it and then we will discuss, so this is the encoding scheme that is followed in genetic algorithm and we will discuss about other operators. The next operator that we will discuss in the next lecture slide is the selection scheme. That means, how to select the best solution out of many solutions are there. So, we will be discussing in the next lecture.

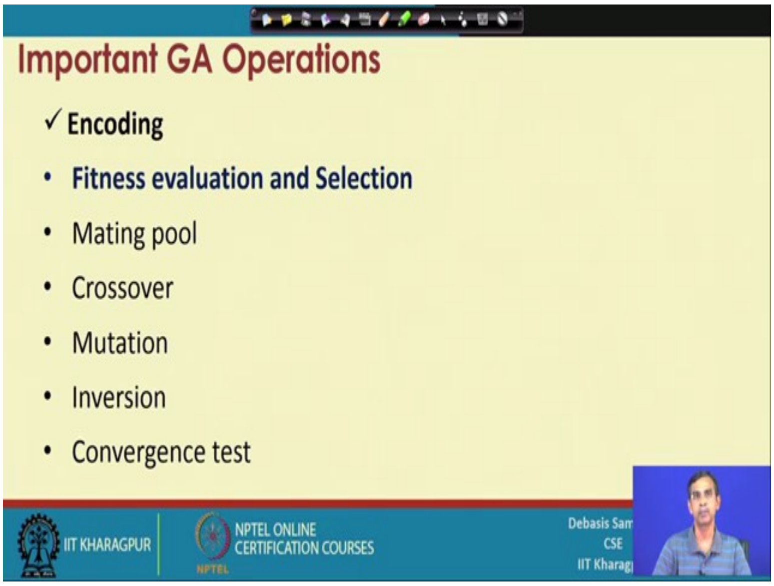
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 18
GA Operator: Selection

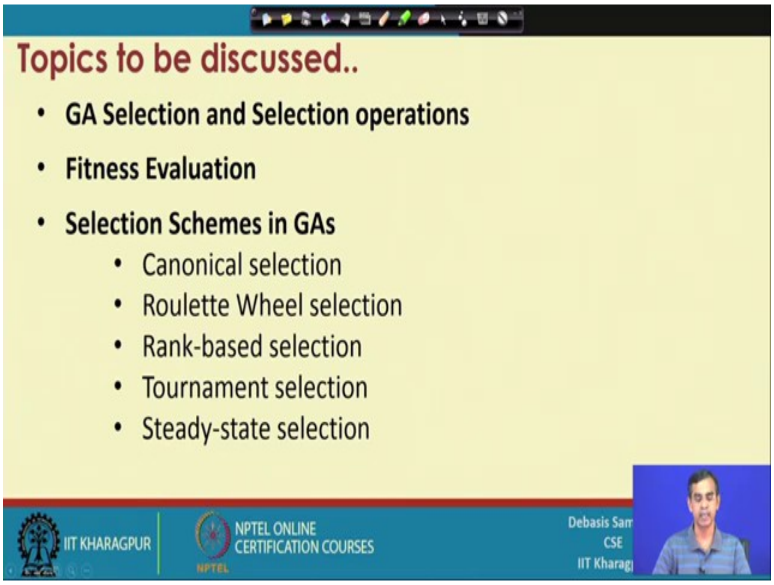
We will learn in this lecture, the another GA operator is the selection operation.

(Refer Slide Time: 00:24)



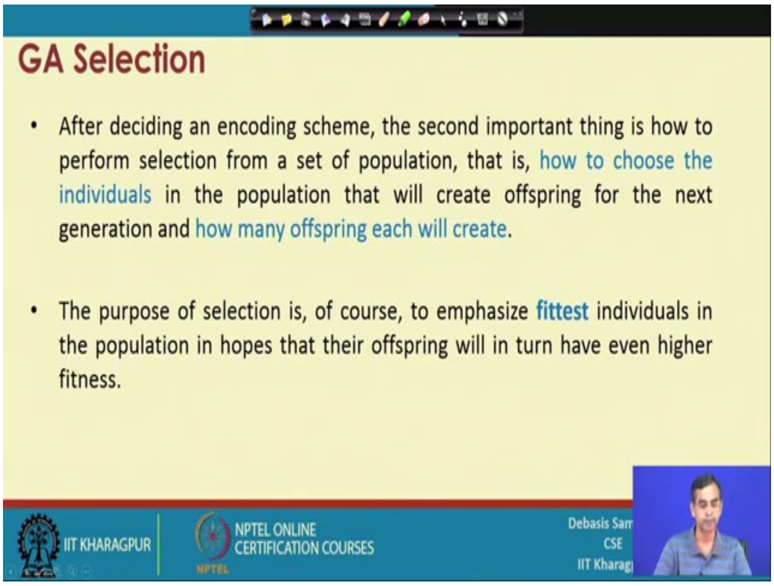
The selection operation is basically the fitness evolution for the solutions that is there in the current population. So, there are different what is called the techniques for the selection mechanism.

(Refer Slide Time: 00:35)



And we will discuss five techniques in this course in this lecture, the five techniques are the canonical selection, then the roulette wheel selection, rank based selection, the tournament selection and steady state selection. So, let us start first with the canonical selection, and then we will discuss one by one.

(Refer Slide Time: 01:02)

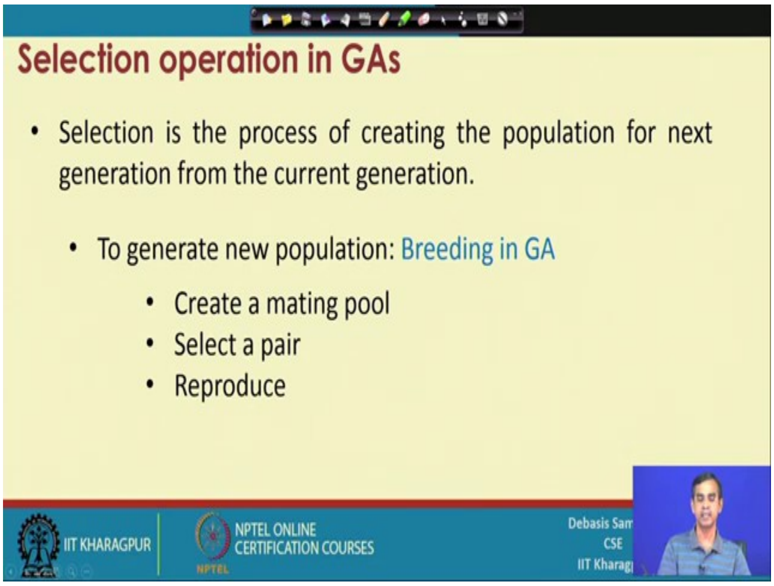


Now, selection is the one important process before going to the convergence test we have to first evaluate the best solution or the fittest solution. So, basically by means of selection our objective is to how to choose the individuals, those are there in the current population at any instant, and then if we choose the best solution then the best solution can be passed through unless if it is not converse; that means, the optimal solution is not achieved then we have to go for the next population and.

So, far going to the next population our task is to select the mating pool. So, usually the procedure of selecting the mating pool is to select the best individual first and then undergo them to the mating and then reproduction. So, the purpose of the selection is therefore, to I mean ensure that the fittest individuals in the current population is selected to produce the better offspring.

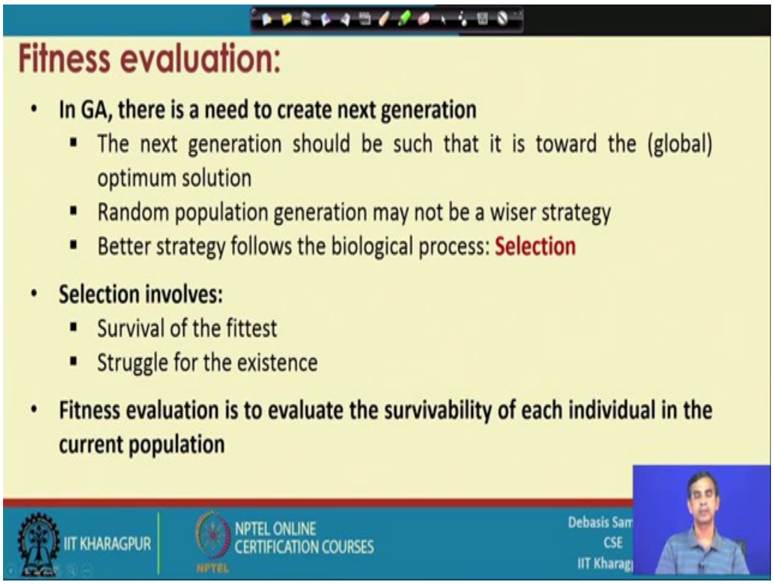
So, this basically necessity the selection procedure is unless the random procedure cannot give the best solution at the quickest way or it will not convert quickly or giving the accurate results. That mean we have to follow certain selection mechanism other than the arbitrary selection or random selection rather.

(Refer Slide Time: 02:27)



Now, so random selection is, so we will this discuss about basically selection. Selection is a prior step of breeding, breeding means reproduction; that means, mating pool creation selecting the mating pair and then reproduction all these things.

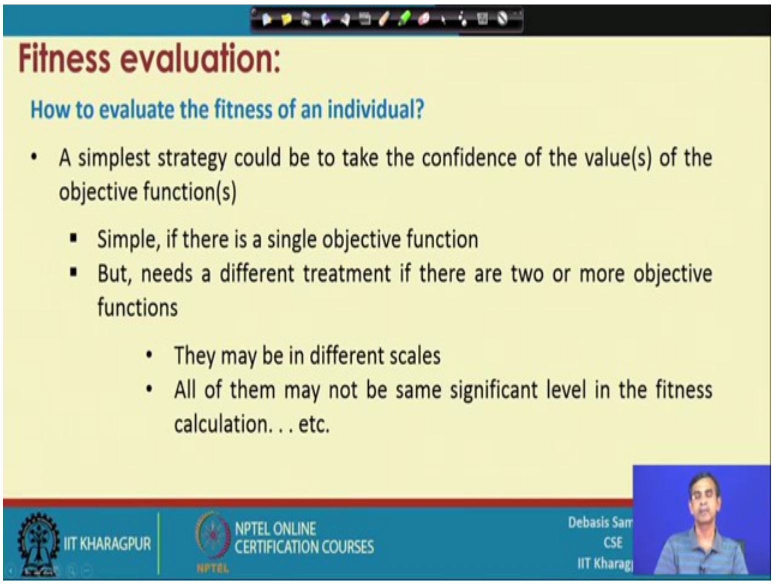
(Refer Slide Time: 02:45)



Now, so far the fitness evolution is concerned. So, as we know so GA, genetic algorithm is an iterative steps. That means, say cyclic process it has to repeat it again and again, or we can say it is basically one population to another population and in each population we have to search for the best solution.

So, best solution is basically ensured by the selection operation in the genetic algorithm and the fitness evolution is the one scheme that will allow us to evaluate the survivability of in each individual in the current population. Now, let see what are the different method are there or the selections are there.

(Refer Slide Time: 03:30)

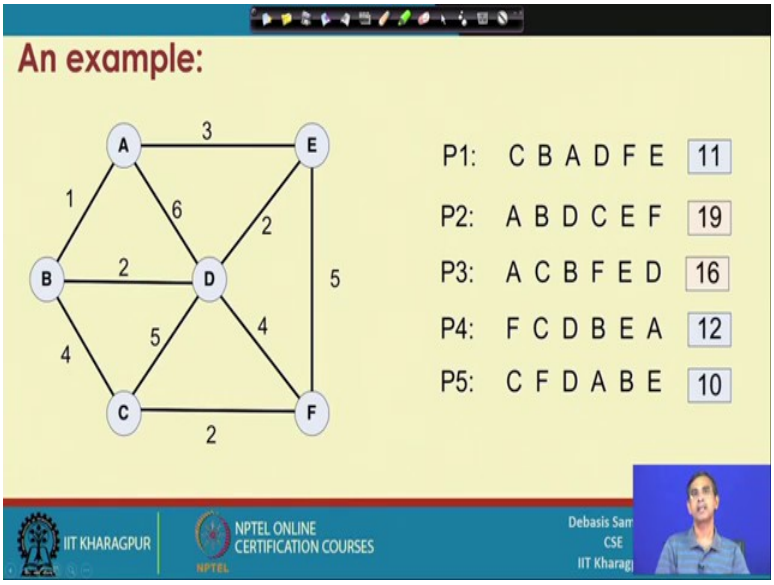


Now, the question is that how to evaluate the fitness of an individual. So, there should be some metric or some policy should be there by which we can apply this one at our hand we have the chromosome or an individual solution and the individual solution is represented by a chromosome we know. So, given the chromosome we have to obtain the fitness value that is the objective actually.

So, idea is basically you know, so at any instant one individual basically represents what are the different values of the design parameter at that instant. That means, if the objective function is f defined in terms of n design parameter, then its phenotype represents that at any, at that instant the different values of this design parameter

Now, so this basically can give us quickly to calculate the objective function. So, one way evaluating the fitness value is basically same way, let see lets calculate the objective function for the current values, of the gene values, and then this can be a fitness value. So, this is basically broad idea that is followed there.

(Refer Slide Time: 04:48)



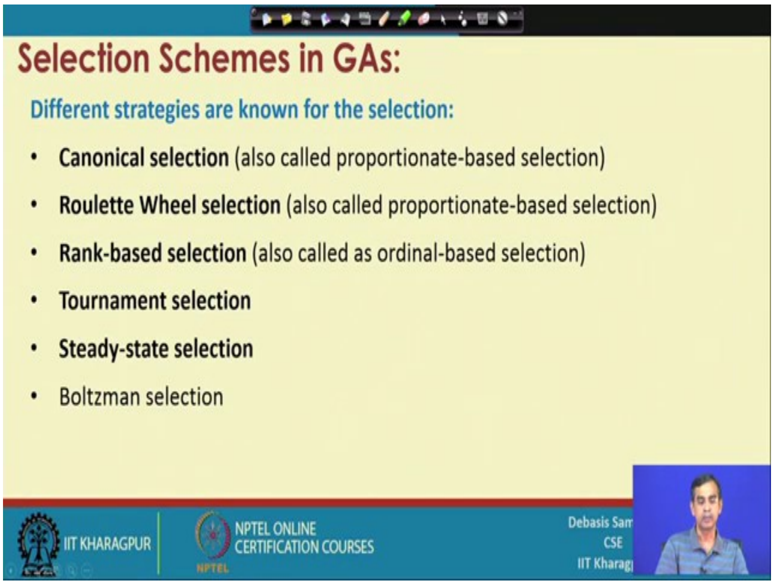
Now, the idea it is again we can represent it like this. So, this is an example to explain how the fitness evolution will work for us. So, this is in the so we can this in the refer with reference to the travelling salesman problem like.

So, here basically five cities are given there A, B, C, D, E six cities, of course, six cities problem A, B, C, D, E, F and these are the different solutions. So, for example, P1 is one path and these are the different solutions are there and the cost of this solutions as we know exactly the cost means; what is the cost of C to B, then B to A, then A to D, then D to E, and D F to F to E, like the cost is basically here in 11.

So, at any instance suppose this is the population. Population includes five different individuals or five solutions, and then we can apply the cost function. So, applying the cost function for P1, it gives this is the cost value and similarly this one.

Now, out of this different cost value that is obtained in this case, we can say this 19 is the best. So that means, out of the solution if it is a minimum cost of course, is not the maximum right. So, if it is a minimum cost then out of the solution 10 is the best solution the next best is 11 and then 12 and like this one. So, 19 and 16 are the worst, two worst solutions here. So, this basically gives an idea about how the fitness is, fitness can be collect can be calculated with the help of the objective function.

(Refer Slide Time: 06:41)

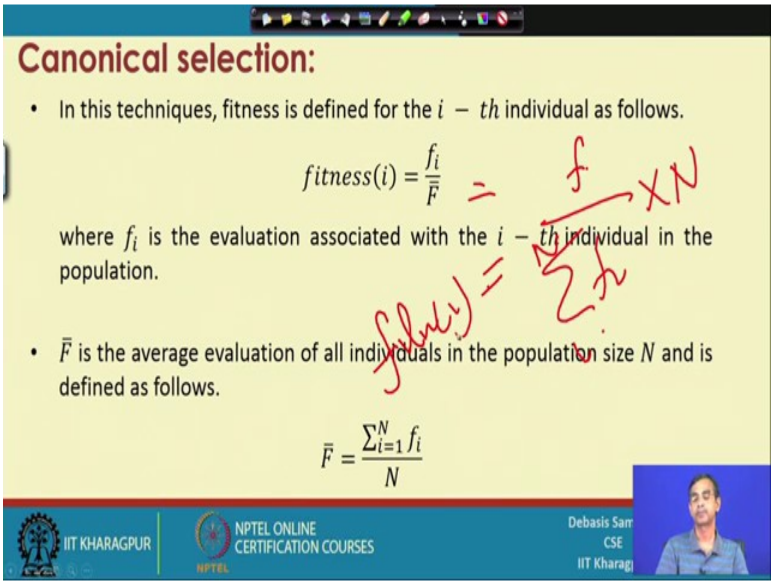


Now, so we can say in other words that fitness value for the measurement of fitness value one metric that can be considered, is a objective function. Usually, it is followed in many of the cases, otherwise it is some other strategies or policies followed.

Now we will discuss about different selection; that means, that is depending on the survivability or some other way or some more procedure how from a set of I mean from a given population, how the best some solutions can be selected for the mating or for the mating pool actually. Now, we will first discuss about the canonical selection the canonical selection. So, first we will decide canonical selection, the in GA theory this canonical selection is also called proportionate based selection.

The like canonical selection the roulette wheel selection is also another version of this it is also called proportionate based selection. The both are the proportional based selection the rank based selection, again it is called the ordinal based selection, and then there are some other selection strategy also known it is called the tournament selection steady state selection anyway we will discuss one by one then, first let us discuss about canonical selection.

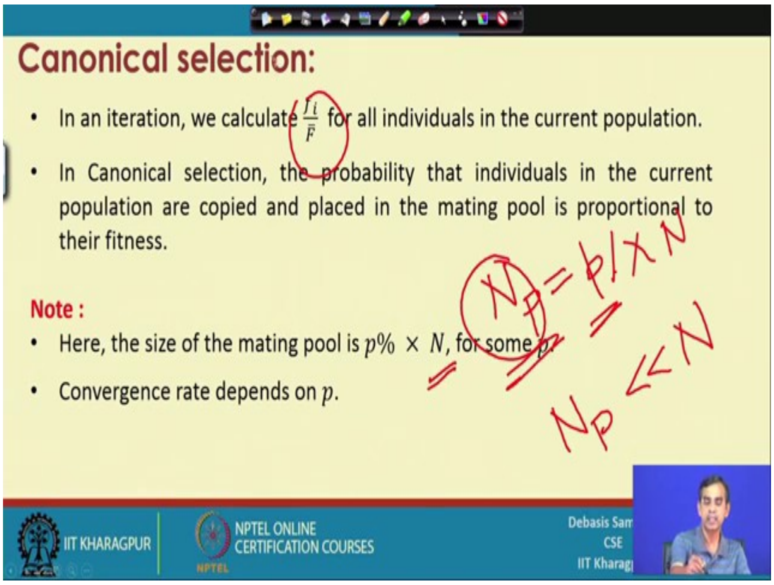
(Refer Slide Time: 08:12)



Now, canonical selection idea is very simple, it basically calculates two values for each individual. So, f_i fitness value of the i-th solution and this $\acute{F}$, this represents the it is represents the average evolution of all individuals in the current population, as in that population of size N then the $\acute{F}$, can be calculated using this formula.

If we know the fitness values of each individual, in the population then it is a summation of the fitness value of each follow divided by the total number of size of the population so this gives the $\acute{F}$. So, essentially, so this is basically it is same also $\frac{f}{\sum_{i=1}^{N} f_i} \times N$, this is the formula that is can be followed to calculate this one. So, these basically calculate the fitness of any i-th individual. Now, the canonical selection follows this formula and we will see exactly how this formula can be applied to I mean to you use the selection of mating pool.

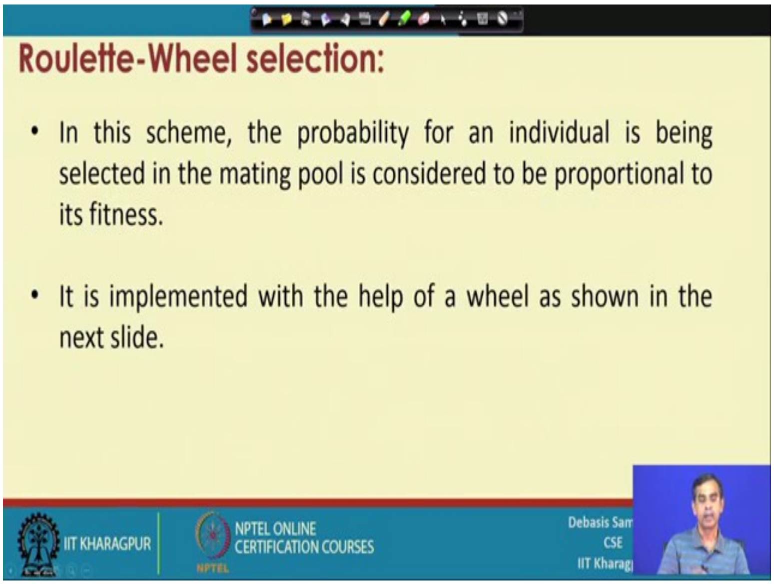
(Refer Slide Time: 09:34)



So, in any instance of the current population, we calculate this fitness value, this fitness value for all individuals and according to this canonical selection, the probability that an individual in the current population will be selected for the mating pool is proportional to this fitness value; that means, the individual which has the highest value of these value will be selected first then the next value will be selected and this will continue to select N_p number of solutions.

So, N_p is basically some p of N. So, p maybe 25 or maybe 30 or like this one; that means, this way out of the N. Where N is the current population size. We will select N_p number of individuals, where $N_p < N$ and then they will be selected based on this fitness value that we have calculated for each that mean the higher the fitness value will be selected first and so on, so on, this way we will select N_p. So, this is the concept of canonical selection here.

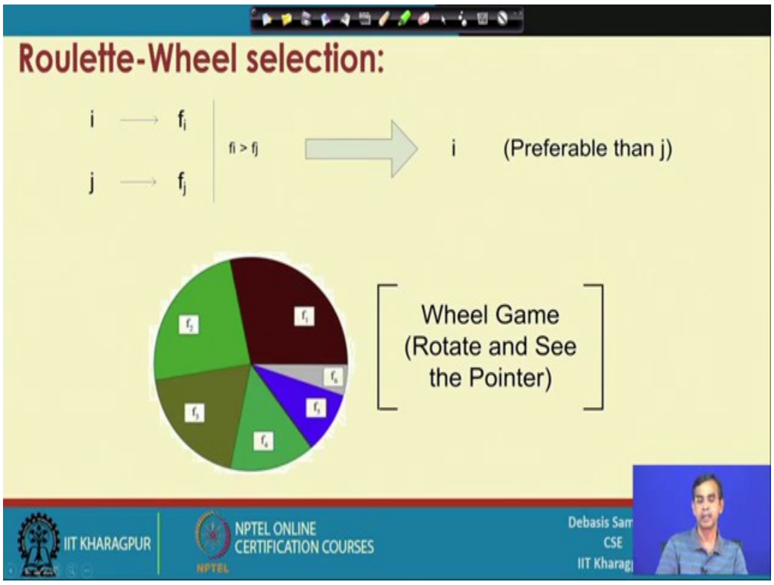
(Refer Slide Time: 11:03)



Now, after knowing the canonical selection, we will learn about the next selection strategy. It is called the roulette wheel selection strategy. In this scheme the probability for an individual is being selected for the mating pool is considered to be the proportional to its fitness. That is basically the concept of canonical same as this one that is why both the technique is called proportionate based, but it has little bit different the idea.

The idea is basically, if we know one wheel let see exactly what is the idea that is followed there in roulette wheel selection it is there.

(Refer Slide Time: 11:44)



So, roulette wheel selection can be better understood by a wheel game. So, wheel game is basically is a wheel like this and the different wheel is marked with the different colour or different symbols and then the different area, rather or different regions, the different regions is proportional the different fitness value.

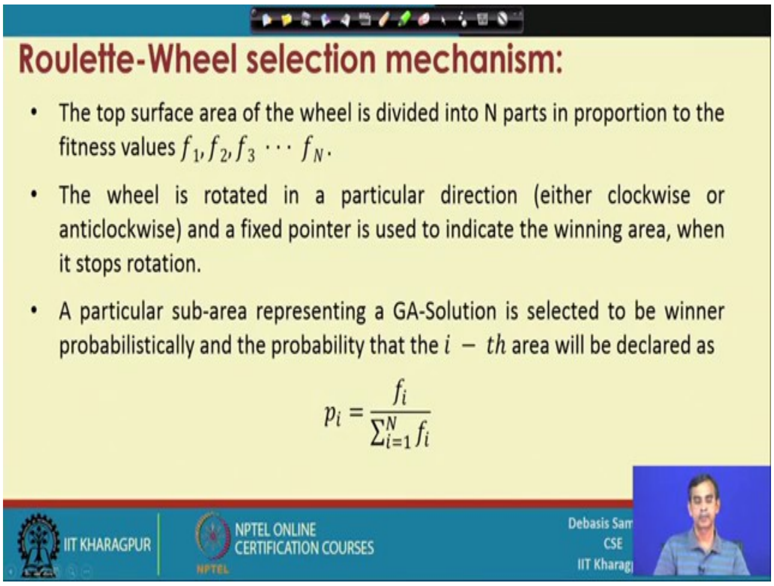
For example, see suppose one solution having the fitness value f_1. And it if it is area that is can be cover proportionally so this is, and similarly, f_2 is this one proportion and this one. Say here basically f_6 is the lowest fitness value for the solution. So, the wheel can be calibrated based on the different fitness values and it is like this one. Now having this is the wheel and supposes it is rotated in this direction and there is a pointer like.

And let it be rotated and then when the wheel stop it, it will point to some point it is there. So, if it points to this one then the f_1 will be selected. So, let the wheel be rotate for N_p times, where N_p number of individuals to be selected for the mating pool. So, for each time we will select the individual which basically this one.

So, the probability that an individual will be selected for this, in fact, proportional to their fitness value become. So, this has the least chance to stop it there because it is the wheel game is like that. So, the idea it is followed there the same idea is basically followed in the wheel scheme is there and then.

So, it is basically the, it can be; obviously, that wheel on that if the fitness value is greater than, then that greater fitness value the individual with greater fitness value will be selected for the mating pool.

(Refer Slide Time: 13:41)

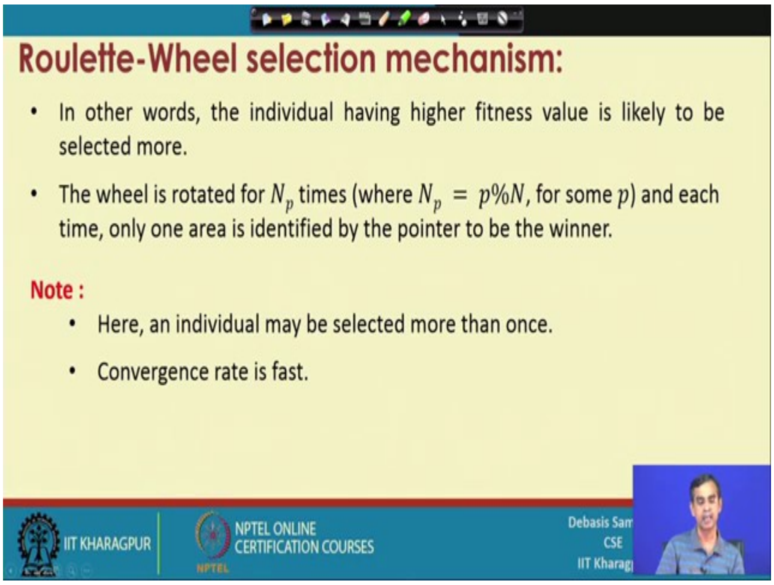


Now, this idea is basically followed there in the roulette wheel scheme. Now, the mechanism is therefore, the top surface area of the wheel is divided into N parts in proportion to the fitness values of each individual. The wheel is rotated in a particular direction may be clockwise or anticlockwise and a fixed pointer is used to indicate the winning area when it stops rotation.

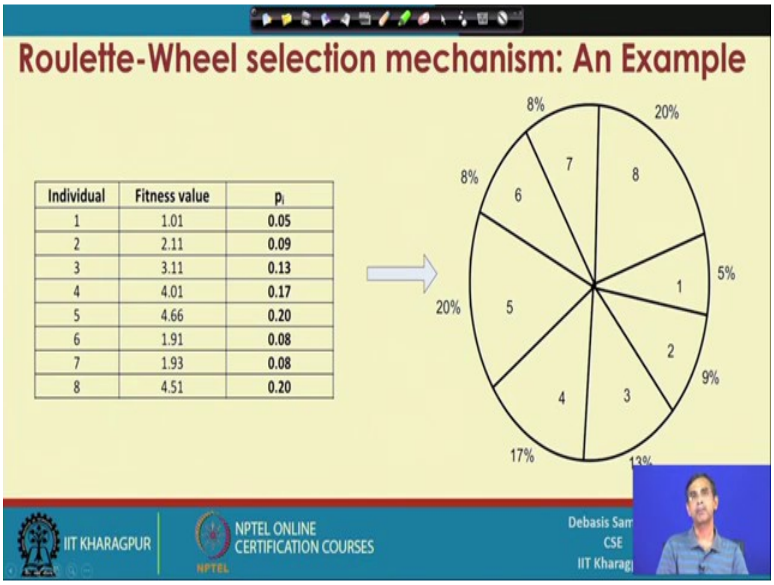
A particular sub-area representing, a solution is selected to be winner probabilistically and probability that the i-th area will be declared as the winner is using this formula. So, this is basically the formula that basically p_i , that this particular area will be selected.

Now you can note that this formula p_i is basically same as the fitness value, that we have calculated in case of the canonical wheel. Only the product n is there, but here it is not there and otherwise it is same so, but in this roulette wheel scheme this basically gives the calculation of the probabilistic value, that at any i-th rotation of the wheel, the i-th individual with the fitness value f_i will be selected for the mating pool. Now, having this is the understanding; then let us see how it will work? It is like this.

(Refer Slide Time: 14:59)



(Refer Slide Time: 15:06)

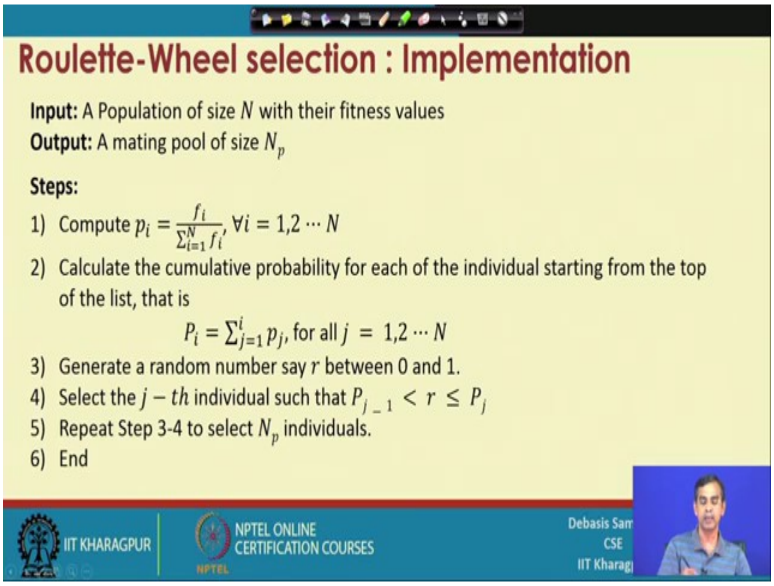


I can give an example, so that you can understand it. Say suppose at any instant there are eight individuals, and each individual has their fitness core which is mentioned here right. These are the fitness core for eight individuals is calculated by some means and then based on this fitness core and using the probabilistic calculation, we can calculate these are the probabilistic value for each individual.

So, for the eight individuals their fitness can be calculated and finally, their probability of selection can be calculated. Now the same thing it can be represented here, so these are the different fitness value of the different solution, and then these are the probabilistic value that is there for each.

So, this basically the, so if this is the roulette wheel alternatively this is also the roulette wheel in the tabular form this is the pictorial form of the roulette wheel and this is the tabular form of the roulette wheel. Now, I will see exactly how roulette wheel mechanism can be followed to select the individual.

(Refer Slide Time: 16:16)

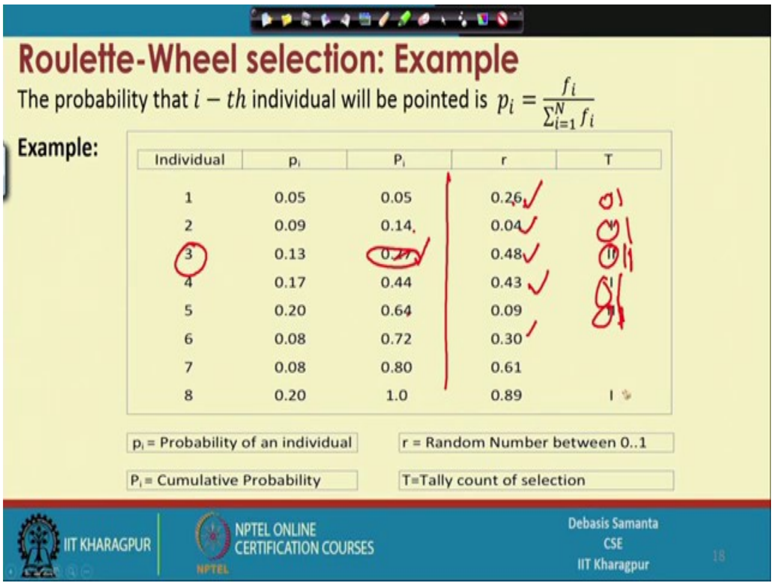


So, there are few steps involved, so as we have said already input is the N number of individuals in a current population, and output is basically we have to select N_p number of population out of N, the for the mating. So, the task, task in this method is compute pi for each, even the f_i the fitness value for each. Then next step is basically calculating the cumulative probability for each.

So, its cumulative probability value can be calculating in this formula. So, $j=1¿i$, p_i for the i-th individual and it is denoted by capital P_i. So, it is basically called cumulative probability. Now then we have to generate a random number between 0 and r. Let, this random number be 0 let this random number be denoted as r. Then if the r is in between this cumulative value P_j and P_{j-1}, then select P_j -th individual as the winner, and we have to repeat these steps right for N times to select N_p number of individuals.

So, the idea it is like this we have to calculate f_i, the fitness core for each individual and then we will calculate the probability of selection p_i, for each individual and then cumulative probability and then generate a random number r and based on the random number r in between 0 and 1, we have to decide the j-th individual for the selection. Now, we can give an example, so that we can understand it better.

(Refer Slide Time: 18:00)



Now, here is a one example like this earlier. So, these eight individuals these are the p_i value the probability of searching is here and then these are the cumulative probability. Now, you can see what is the cumulative probability is basically, for the first it gives 0.1 then this is added to give this one, then this is added to this one it is give this one.

So, this way so it is basically the so 27 means, it will add this one, then this 27. Similarly, this one means adds up to this one this is the cumulative probability this one. So, this way the cumulative probability for each will be calculated. Now, if eight individuals are there for their p_i, p_i and then P_i can be calculated. So, this is the step next is the selection according the roulette will.

So, idea is that we have to select at random number in between 0 and 1. So, let us see this is the current instant the first toss and the random number is 0.26. Now if so then we have to select the winner out of which how you can select it. So, if the 0.26 is less than to the highest cumulative probability and then. So, 0.26 is basically lie in this one, so 0.26 I mean; that means, it will. So, it is basically this value should be less than the cumulative value greater than this one.

So, it is basically this one is the selected that means; we will select 3. So, if 0.26 is tossed first then the third individual that mean it will be selected. So, it gives a tile mark here 1 so this is selected.

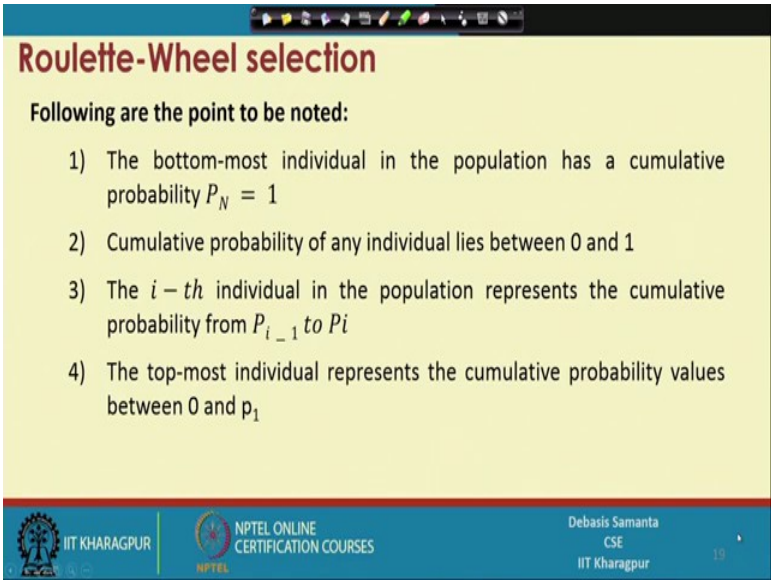
Now, next is 0.4, so 0.4 is basically selecting this one because, it is within less than this one. So, one will be selected, so 1 will be tiled here, then 0.48 the next toss now 0.48 is basically this one. So, it means select 5, so we can be tile one.

Then 0.43, 0.43 means four. So, the 4 element will be selected; that means, this will be selected. Then 0.09, 0.09 will be this one. So, 2 will be selected, then 0.30, 0.30 is this one, so 4 will be selected. So, 0.30, so 0.30 under this category, so 3 will be selected, 3, 1 selected first then next 3 again selected, and then 0.61, 0.61 is basically. So, five will be selected, so 5 is selected earlier 5 selected next 5 are selected second times, and 0.89 it basically 8 will be selected so 8 will be selected.

So, this tally mark basically shows the how many individuals are selected and if an individual selected more than once or not it is like this. So, in this example we see the individual one is selected once two is selected once. However, these selected three I mean the individual 3 selected twice, fourth is 1 and then individual 5 is selected twice; however, individual 6 and 7 never selected at all and then individual 8 is selected once. So, these are way so 8 here basically, eight round we have carried out an in eighth round how the eight individual will be selected there.

So, we need not to carry the eight rounds. In fact, we have to carried out N_p rounds depending on how many individuals you have to select for the mating pool actually. So, this procedure needs to be repeated for N_p times and this time we have to have a random number in between 0 and 1 and accordingly we have to select it. So, this basically the idea about the roulette wheel scheme and then we will follow it there.

(Refer Slide Time: 21:58)

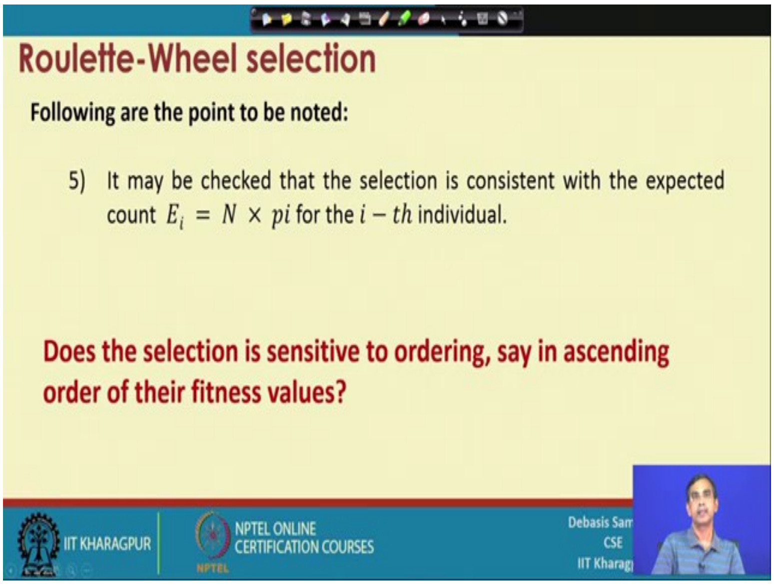


Now, following at the important features in the roulette wheel scheme we can understand it the bottommost individual in the population has a cumulative probability $P_N = 1$. So, this can be example like this so it is the bottommost means this is the bottommost individual and it has the cumulative probability 1.

The cumulative probability of any individual lies in between 0 and 1. So, we can see again the cumulative probability always in between 0 and 1. The i-th individual in the population represents the cumulative probability from p_{i-1} to p_i. So, it can be like this, so the i-th individual if suppose the fourth individual as the cumulative value between p_i to p_{i-1} to p_i that means, third to fourth cumulative probability that is the problem that it will be selected depending on the random number that can be.

And then the topmost individuals, in this representation having the cumulative probability values 0 and then first p_i. So, again this is the topmost one in this top most the cumulative probability in between 0 and 0.5 that is basically the first one, so these are this one. So, these are the property that the cumulative probability holds in this case.

(Refer Slide Time: 23:35)



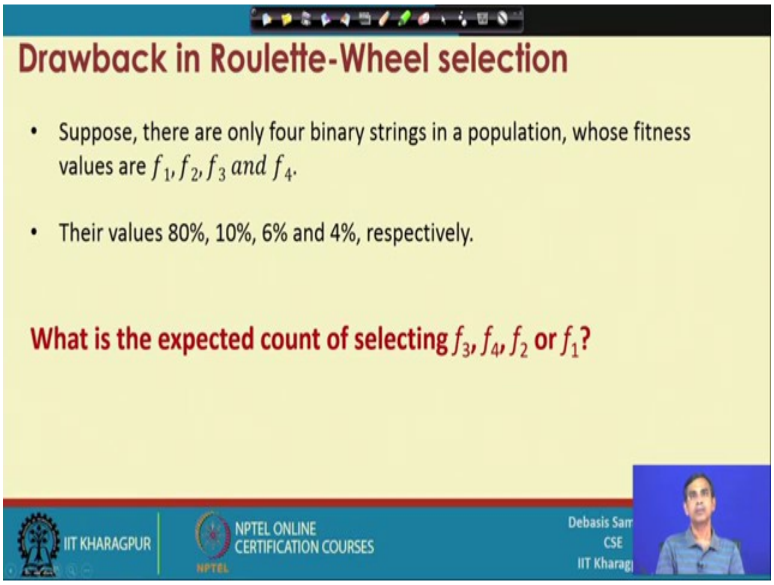
And one idea it is just I have to say it that the expected count that can be obtained that can be also proportional to the probability that the i-th element will be selected. So, it basically holds good this formula if E_i denote, that the how many times that E_i will be selected expected for the selection that can be calculated.

If N is the size of the population and p_i denotes the probability that i-th individual will be calculated, then it gives the expected selection. Now so, this idea is basically, this basically the idea about the roulette wheel scheme rather. So, in the roulette wheel scheme ok.

So, that tally mark that we have used is basically counting the expected count for each individual actually. So, it is basically $N \times p_i$. Now here obviously, one question that arises is that whether this selection is sensitive to ordering say in ascending order of their fitness values if we do of this one.

What I want to say that it is independent of the sensitivity of the ordering, if you take the individuals in any order it will give the results in a probabilistic manner of course, but it is independent of the any ordering scheme that we follow. So, whether all the individuals are ordered according the fitness value and then their p_i value is calculated and then cumulative probability it hardly matters.

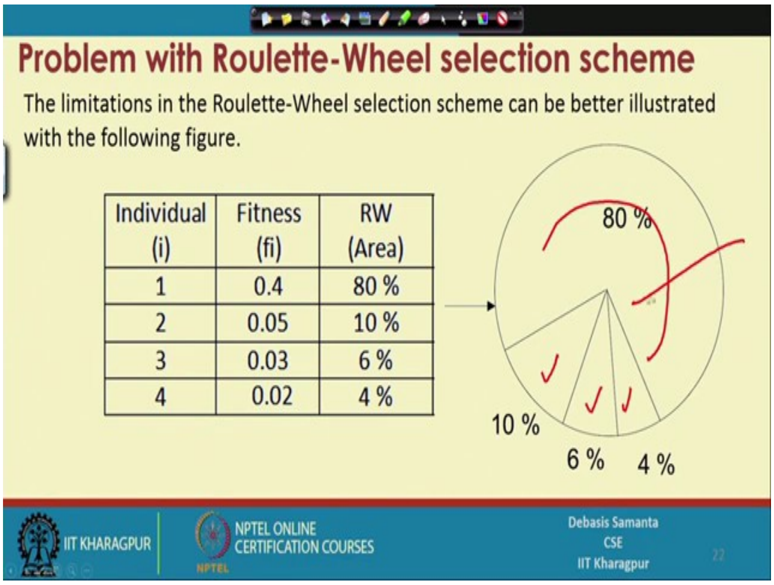
(Refer Slide Time: 25:14)



Now, so roulette wheel selection is more, I mean a better approach than canonical selection because canonical selection is basically very naive approach, but roulette selection gives a favour. However, it suffers from one limitation which I want to discuss it as an example the limitation of the roulette wheel can be understood. Suppose at any instant there are only four individuals and they are denoted by $f_1, f_2, f_3, \wedge f_4$, and their fitness value is here.

So, 80, 10, 6, and 4, these are the fitness score in percentage we have represented it anyway. So, it is there or we can say 0.8, 0.1, 0.6 and 0.4 are the fitness score of all these things on there. Then if we apply the roulette wheel scheme, then let us see what will happen to these kind of selection and these basically gives an idea about what is the drawback of the method.

(Refer Slide Time: 26:08)



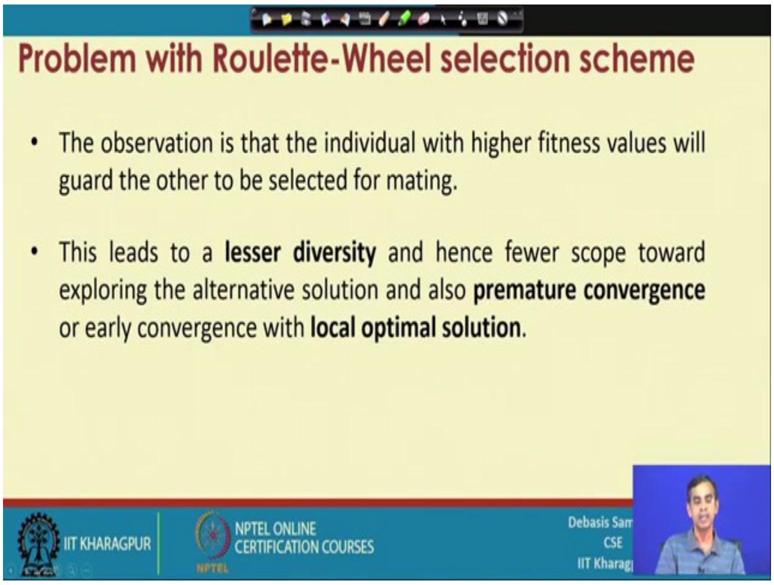
So, it is basically the fitness value of the four individual that we have used it, and then these are fitness score we have represented it in the point formula it is like this basically is the same 80 to this one also same actually the representation.

So, if we see the roulette marking wheel game then the individuals covered with the highest one. And then this is the second individual, third individual and fourth individual, and if we play the game then the chance that the maximum time it will point is the individual one. So, if we run it for the four times the probability that the first individual will be selected. That means, the wheels will favour that individual which having the fitness value it is desirable.

But sometimes it is also not, so desirable because it deprives or it basically ignore the other to be selected, and in the GA strategy the idea is that you have to give a fair chance to all other also to be selected. Obviously, the best individual will be selected most of the time, but sometime the other individuals if it is selected it can give the better diversity in the problem solution.

So, we should not ignore the other individuals also or other that always favouring to a particular individual also not a good what is called the strategy to follow.

(Refer Slide Time: 27:36)



So, that is why this is idea is that the higher individuals with, the individuals with the higher fitness value will favour according to the roulette wheel mechanism and these become a problem or it creates a lesser diversity and hence there is a chance that the genetic algorithm will terminate with a local optimum solution or the premature convergence will resolved.

So, this is the one limitation there in this roulette wheel selection scheme and the same limitations can be overcome using some other approach and this approach is called the roulette selection scheme, and the roulette selection scheme will be discussed in our next slides next lectures.

Thank you.

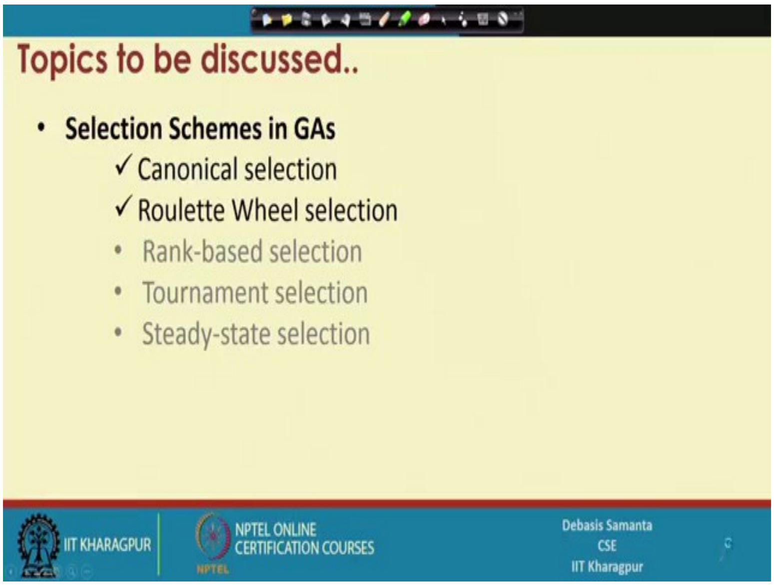
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 19
GA Operator : Selection (Contd.)

We are discussing the selection operation in genetic algorithm. Today, we will discuss the few more applications, few more selection operations. So, the in the last lecture we have discussed about two selection operations, namely canonical selection and roulette wheel selection, we have learned there are certain limitation in roulette wheel selection.

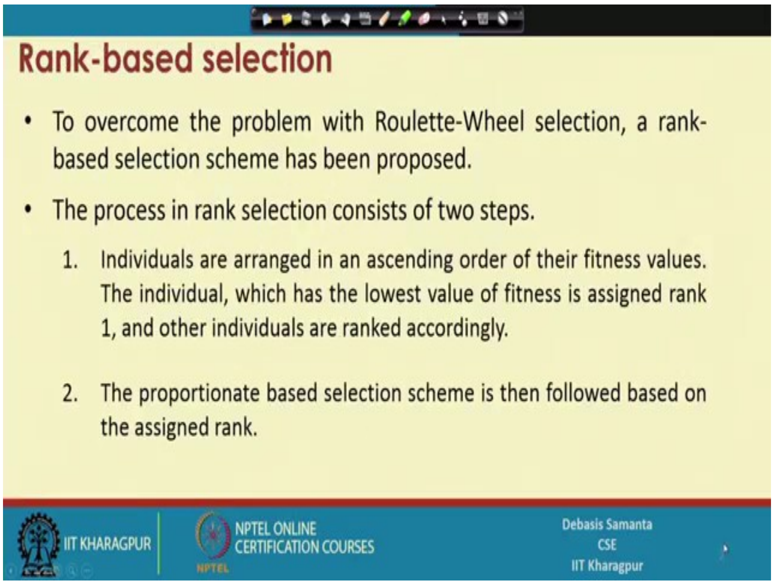
The other selection operations that we are going to discuss it today, these are rank based selection the tournament selection and steady state selection.

(Refer Slide Time: 00:56)



So, let us first discuss about rank based selection.

(Refer Slide Time: 00:56)



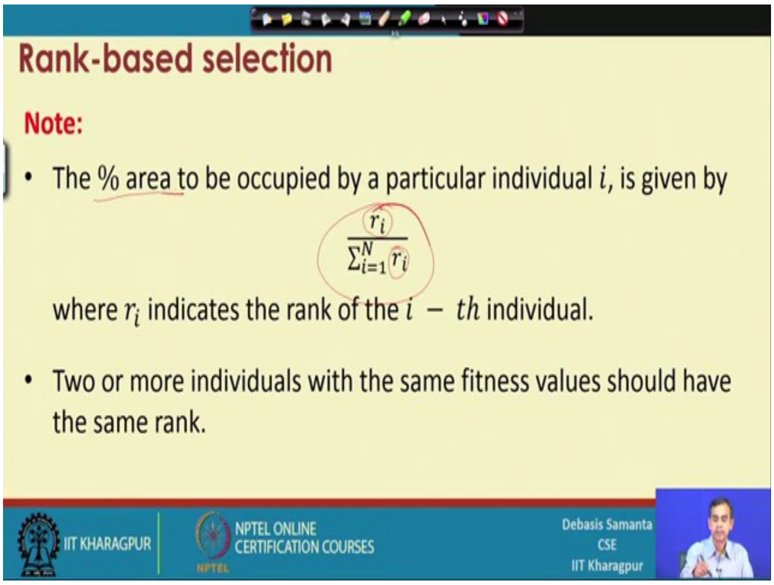
The rank based selections have been proposed to address the problem that is there in the roulette wheel selection. We know that in the roulette wheel selection basically the individuals, which has the highest fitness value is preferred more compared to the lowest fitness value.

Sometime this is not a fair selection rather we should give a chance to select the inferior population also, because sometimes the offspring that will be produced by the mating of the best solution and then worst solution can lead to a better solution with faster termination. Anyway, so rank selection is basically best the basically proposed to remove the biasness towards the high fitness value populations. So, in this process these processes consist of brought two steps.

So, in this process all the individuals are arranged in an ascending order of their fitness values, and then the individual which has the lowest value of the fitness is assigned as rank 1 and then the next lowest fitness value is rank 2 and so on.

Once the rank is assigned to each individual then we follow the any proportionate of the selection scheme likes a canonical or roulette wheel then, and this way we will see how the favour towards the highest fitness value that is given by the roulette wheel have been checked in rank based selection method.

(Refer Slide Time: 02:50)

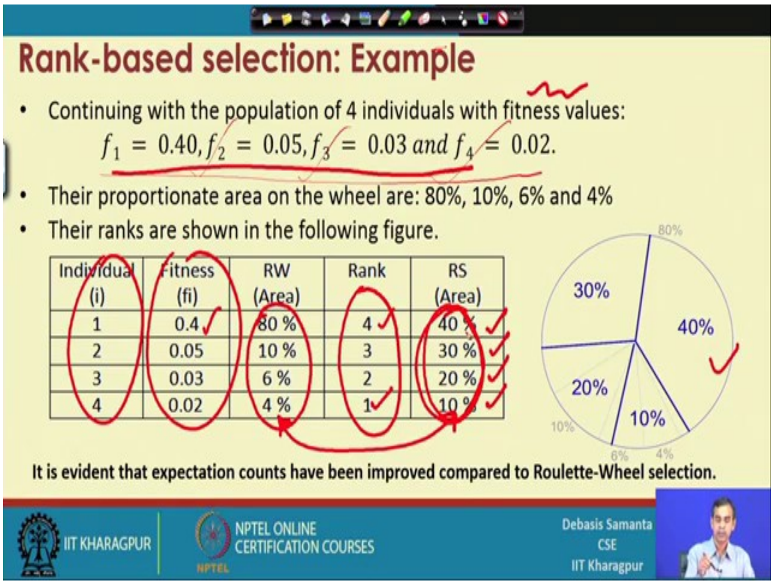


Now, so the rank that will be assigned it basically decides the percentage area to be occupied by a particular individual say it is i and the formula that we follow to for this purpose is it is written here.

So, it is the same as the fitness value or fitness core calculation that we have learned in roulette wheel scheme, there we basically r_i was replaced by f_i there and so there. And this basically denoted as a p_i in the roulette wheel selection strategy, but here it is the calculation of the rank and based on this rank, we decide the probability that it will be selected for the things. Now, here the percentage of area basically indicates as in the roulette wheel. That means, it is the proportionate that it will be I mean it is eligible to select for the next generation.

Now here there may be sometimes that i ; that means, two or more individuals with the same fitness value is the quite possible, in that case we should assign the same rank to the individuals ok.

(Refer Slide Time: 04:20)



So, after assigning the rank our next task is to up is to follow, either roulette wheel method or canonical selection method, but not based on the p_i or fitness there rather it is based on these r_i ; that means, the rank values now let us illustrate the rank based selection scheme with an example and in. So, let us consider only the four individuals at any instant, and they are $f_1, f_2, f_3, \wedge f_4$ and the value is shown here, $f_1, f_2, f_3, \wedge f_4$. then ok.

Let us illustrate the rank based selection with an example, in this example we consider four individuals, here as we have shown the four individuals here $f_1 ¿ f_4$ with their different fitness values, and the same populations are listed here the four individuals and their fitness values listed there. And so these are the rank that we have assigned so the individual with the lowest fitness is here. So, the rank is 1 the highest here the rank is 4 and so on, so on.

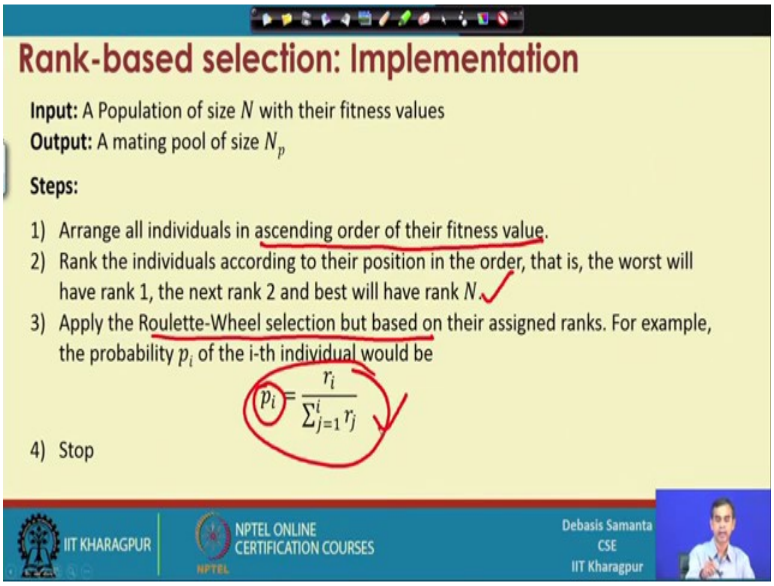
So, this is the rank assigned for the each individual and then on the basis of rank and this is basically the calculation according the rank selection method about their score. So, it is basically point 4, point 3, point 2, point 1, is basically rank divided by some of all the rank like so these are the things are there.

Now, we will apply the proportionate based any selection say roulette wheel on these values there. So, typically it is basically like, look like this one, now if we follow the same thing, but only without any rank then roulette wheel selection will assume these are the values, so the difference is in there right.

So, roulette wheel selection follow these are the course and whereas, the rank selection follows these are the course. So, we can see the 80%, which is considered in the roulette wheel it become consider a 40% at the score and, so on, so on. So, the score has been little bit changed because of this rank calculation and then the rank based selection strategy.

So, this way we can give some favours to the worst individual event for example, earlier which was has given the weightage 4%, it is basically 10% and then also favour towards the highly feet values also reduce 80% to 40%. So, this is the mechanism here it is followed, and if we follow this one the fair chance that the inferior quality, inferior individuals also will be selected compared to the superior in individuals as it is always there in the roulette wheel scheme.

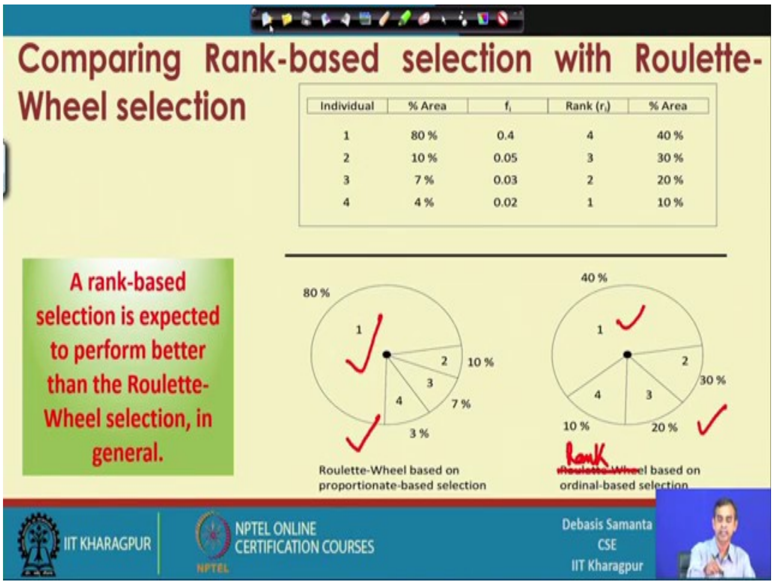
(Refer Slide Time: 07:23)



So, this is the basic concept of the roulette wheel scheme. And so for the implementation of this algorithm is concerned we can explain it very briefly. So, the first step is as we have learned about the first step is arrange all the individuals in ascending order of their fitness values, then rank the individuals according to the position in the order that is the worst will have the rank 1 and the next will rank 2 and then best will rank n so on.

Then once the rank is assigned we have to follow the roulette wheel selection, but based on their assigned rank value which will be calculated like this one. So, basically p_i you can note it the p_i was calculated in roulette in a different manner, but here it is calculated based on their rank, so this is the procedure that we can follow.

(Refer Slide Time: 08:16)

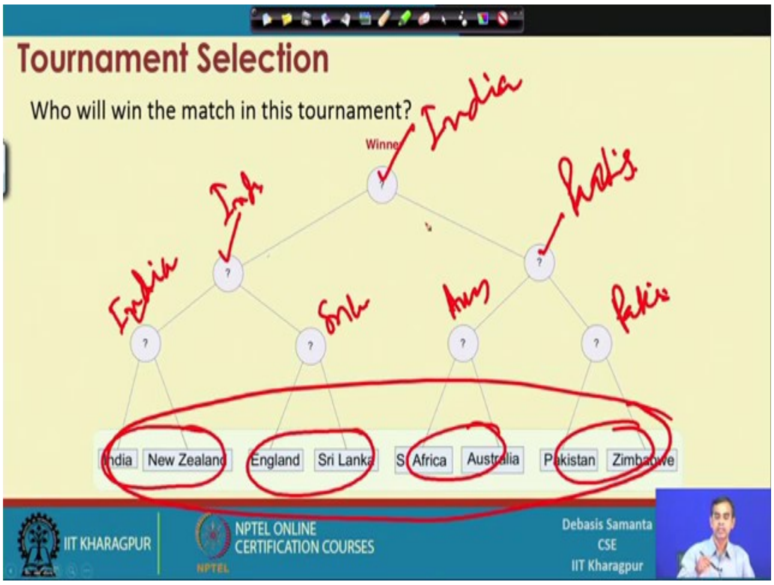


Now, let see how it is beneficial compared to the wheel, roulette wheel based selection. So, the idea it is basically if we follow the roulette wheel selection the wheel game is like this and if you follow the rank based selection the wheel game is like this so it is basically rank based selection ok.

So, in this, so we can say that for the roulette wheel the fitness value having 80%, has the most probable that it will be selected whereas, the probability is reduced here as the area is reduced here.

So, this is why in general we can say that any rank based selection is expected to perform better compared to the roulette wheel based selection in general it is the result that can be obtained, and then it is proved that usually the rank based selection is better selection strategy than the roulette wheel selection.

(Refer Slide Time: 09:15)



Next, we will discuss about another selection strategy. It is totally different than the selection strategy that we have discussed those are the basically based on the proportional based algorithm. Here we have we are going to discuss a new strategy based on the tournament strategy.

Now, so we know the tournament strategy, here there are different type of tournament that can be played, I have given an example here it is basically the knockout tournament. So, in case of knockout tournament the idea it is like this the different games are planned for example, here is the games between the two, what is called the teams is a another, another and different games are there.

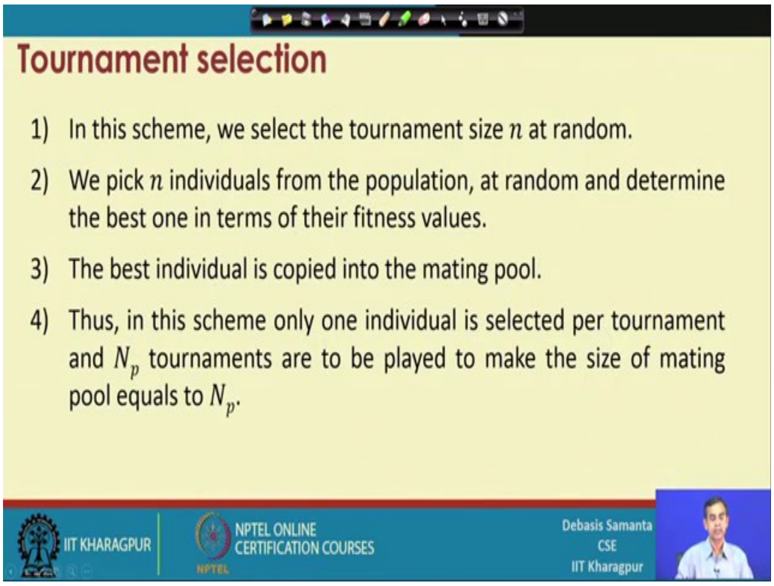
Now, so many games out of the different what is called the players are selected then. So, a game will be played for between the two players here. So, India and New Zealand and who is the fittest here will be selected here, so say India is the fittest India is selected.

Similarly, another play will be played here and Sri Lanka and England. So, Sri Lanka is selected as a Sri Lanka is most fitted suppose compared to the England then the another play. So, this way similarly here also Australia and here is the Pakistan is selected because of some fitness value there.

Now, again another game is played, and so the here India versus Sri Lanka suppose India is the feast. So, in this game India win this and here in between Australia and Pakistan. So, in these game say Pakistan is more fittest than the Australia. So, Pakistan is selected and finally, the play between India and Pakistan. So, here the winner say India is more fittest than Pakistan, so India is selected.

So, this way if we play the game among the different players here and then after the end of the tournament the best, I mean the best the player with the best fitness values will be selected. So, this is the general procedure of the tournament selection strategy, the similar strategy has been followed in our GA, operation also we will discuss this tournament strategy in GA.

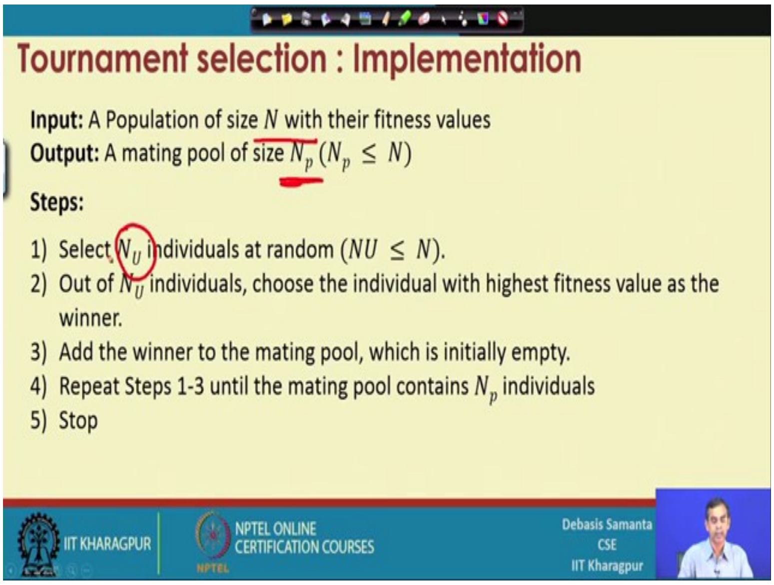
(Refer Slide Time: 11:27)



And that is called the tournament selection procedure. Now, so the GA tournament selection procedure has few steps here we have shown the four steps in this scheme we select arbiter is the team size N at random; that means, say suppose population size is 100, let us decide the team size is 10 say then out of this 10 team.

We have to play the tournament as we have discussed earlier knockout like and then we have to select the one individual and then if we repeat this kind of tournaments for N_p times, where N_p number of population to be selected for the mating pool then we will select this one. So, the idea it is like this.

(Refer Slide Time: 12:09)

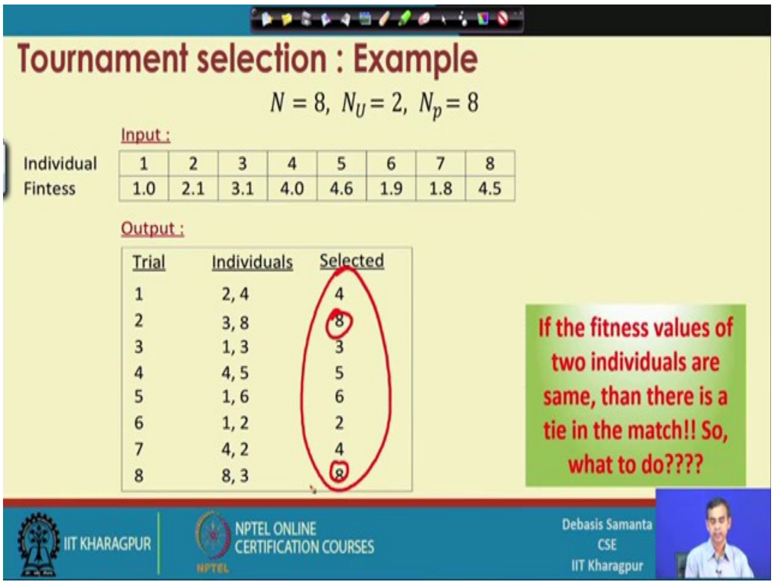


Now, so this is the idea and that algorithm to this idea can be stated like this. So, input is a population of size N , and their fitness values are calculated for each individual and the output should be a mating pool of size N_p , where N_p is some values ¿ N .

So, according the strategy we have to select nu individuals at random. So, you can select this N_U number individual at random and N_U should be very large, very small value compared to N . The total size of the population once the any individual is selected out of any individuals we have to play the tournament knockout tournament like, and then the individuals with the best fitness values will be selected as the winner.

So, the winner which has been selected will be added to the meeting pool which is initially empty. Then we will repeat this step 1 to 3, for N_p times, until N_p individuals are selected for the mating. So, this is the straightforward procedure, in fact, only the calling procedure here in the step 2 which basically plays a tournament among the N_U individuals, but this is also not, so costly and it is manageable and affordable.

(Refer Slide Time: 13:36)



So, this is the I mean strategy that we have discussed. Let us illustrate this strategy with an example and. So, here in this example we can see N the total 8 number of individuals are there and we have to select say, 8 individuals for the mating pool; that means, $N_p = 8$ and N is also 8 here.

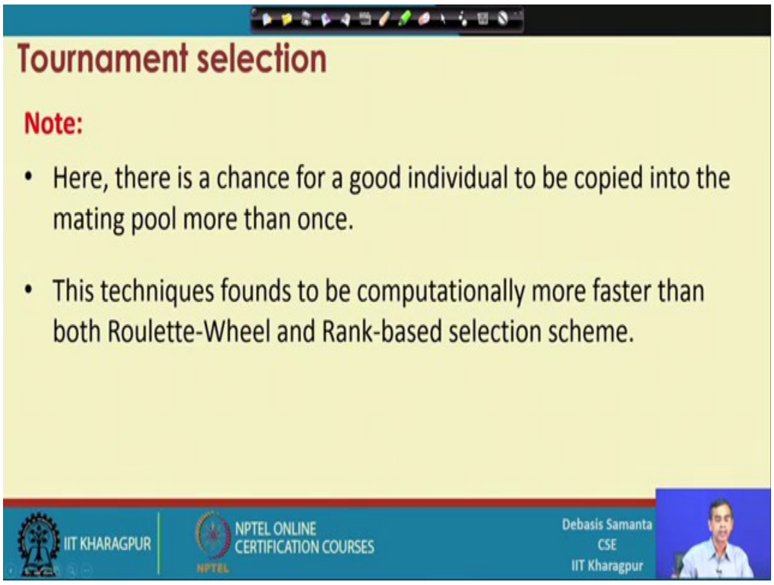
Now, we select N_U size of the tournament is 2 for simplicity and then illustration purpose. So, here the first trail; that means, we select first n individuals from this one at random. So, selected 2 and 4 is selected and out of these 2 two and 4 the winner is 4 because it has the highest fitness value compared to these one.

So, 4 is selected, so this is the first trail similarly second trail 3 and 8 is selected and out of these 3 one 4 point 5 having the high so 8 is selected. So, this procedure is continued and these are the 8 difference select say population that can be selected for the mating pool, here we have intentionally used N and N_p are the same values.

So, here we can see that according to this procedure, the same individual may be selected more than once. For example, here the individual 8 is selected here and here more than once. So, it is quite possible that it can be selected the same things more than once.

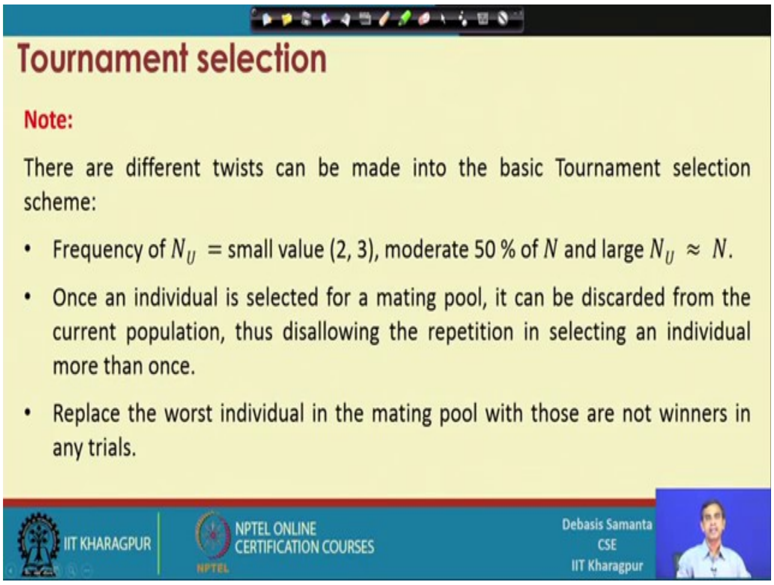
Now, again here for example, there may be another twist another what is called the tie also suppose two individual selected although all individuals are having the same value say suppose 6 and 7, suppose any two individual say 4 and 5, it has the value is 4.0 right, and in a trial if we select the individual 4 and 5 for the tournament for example, here 4 and 5 in the tournament then both have the same value then which I mean individual needs to be selected, so whether 4 or 5. So, in case of tie we can break the tie of certain tossing mechanical and then anyone at random can be selected ok.

(Refer Slide Time: 15:51)



So, this is the tournament selection strategy, and in the tournament selection strategy what we have learned is that, there is a chance for a good individual to be selected into the mating pool more than once that we have already mentioned and this technique founds to be computationally more faster than both roulette, wheel and rank based selection scheme and it also has many other benefits or the benefits of the different selection strategy and their limitations will be discussed at the end of the lectures today. So, this is the tournament selection strategy.

(Refer Slide Time: 16:25)

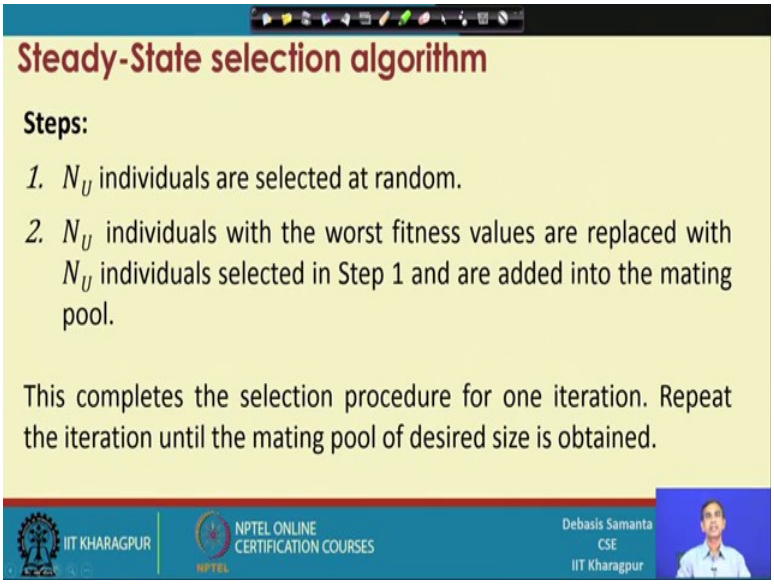


And tournament selection strategy also has certain twist we can follow to make it more applicable and appealing. So, there are few twist which have been hinted here, so the N_U can be changed any value as small as 2 and then as large as N . So, so accordingly the different results can be obtained and then the programmer has to choose which results is favours to give the best result, and then at the fastest execution of the genetic algorithm.

And then another twist can be, so in that case it is repetition is allowed we can check this repetition, this means once an individual is selected for a mating pool it can be discarded from the current population thus disallowing the repetition in selecting the individual for more than once.

So, that is also possible if we want that only one individual will be selected only once and another twist also can be added there in this strategy, if we replace the worst individual in the meeting pool with those are not winners in any trail. So, the worst individual can be replaced by some individuals in the mating pool which are not winners in the trail. So, these are different twist can be followed to make the tournaments selection more robust and more reliable in a different GA execution.

(Refer Slide Time: 17:44)



Next we are going to discuss the another strategy, it is called steady state selection algorithm it is a very simple most strategy it works sometimes very effectively, but not always, but it has computability is very fast and then result is also more or less acceptable compared to the other strategy. So, some programmer follows this kind of strategy also.

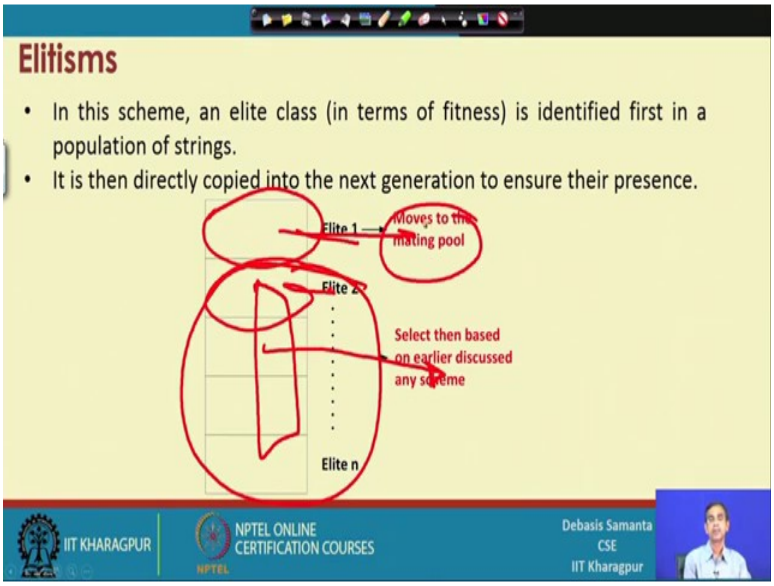
Now, this strategy is basically like tournament selection it also selects one parameter call N_U that is the numbers that can be selected. So, nu can be as small as to usually it is very small number compared to the number of size of the population. So, N_U individuals are selected and they are selected at random. So, randomly any two individuals are selected from the current population.

N_U individuals with a worst fitness values are replaced with the newly selected any individuals in the previous state and added to the mating pool. So, this procedure is repeated for N_p times, where N_p is the number of mating I mean populations to be selected for the mating. So, it is the procedure is like this and if you see as the number of iterations is increase it always refine the current population by replacing the worst.

So, worst will be I from there and then all the based population will be selected and that is giving more fair chances to the individuals with higher fitness value it is selected here compared to the worst one.

So, this is the procedure it is there, it is the simple most procedure one, but it is not so much what is called the control.

(Refer Slide Time: 19:28)



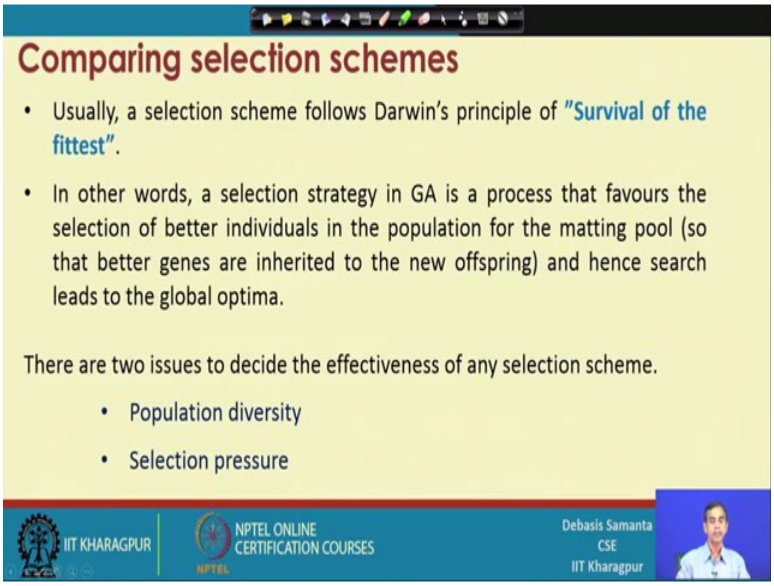
As it is possible there in other selection strategy, but as a simple most strategy it can be followed in some application now. So, our objective is basically to select the individuals which can be played for the mating pool, and we have discussed the four different strategies, the canonical, roulette wheel and then rank based selection. whether the population based and then finally, to selection of the tournament and steady state. So, five selection strategies that we have discussed it.

Now, So, there are again few what is called modifications in the selection mechanism is followed in recent GAs, and this one such mechanism is called elitism what is the meaning idea is that be depending on the fitness values of the individuals the individuals are grouped into a number of elite group, so Elite 1, Elite 2, elite n, so on, so on. So, here basically the idea is that a set of individual based on their fitness value belong to the Elite 1 then other is Elite 2 and, so on, so on.

In other words, the individuals which belong to the Elite 1, they are highly fit what is called population or highly fit individuals. So, the strategy according the elitism that we have to move all the individuals which belongs to Elite 1 to the mating pool, then the then we will select the remaining in order to make the size of the mating pool as N_p . So, select them from the remaining Elite groups, but following some strategy either the roulette wheel or rank based selection or tournament selection there.

So, this will be selected whatever the existing strategy to select the rest of the things and then passed without any selection they are moved to the mating pool. So, this is the one strategy that is followed and it observe that this strategy also very much effective in some situation.

(Refer Slide Time: 21:35)

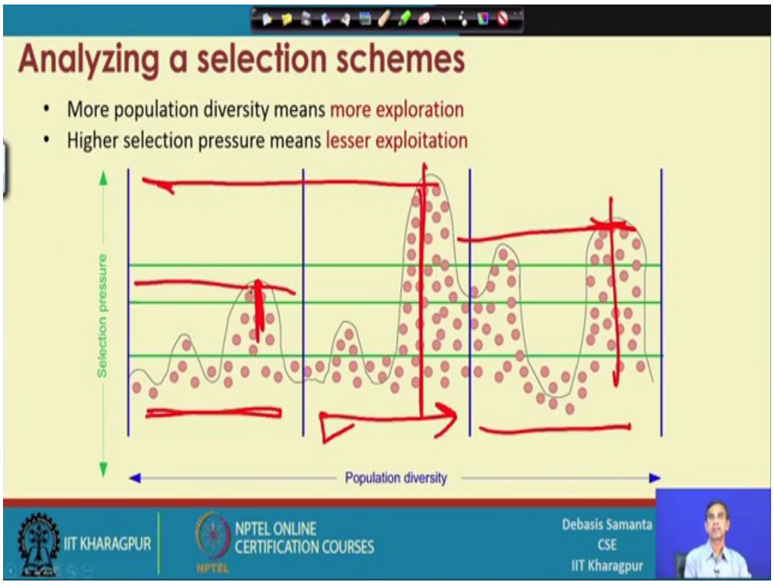


Now, in in short we will compare the difference selections scheme. So, basically the selection scheme follows the Darwins principle of survival of the fittest; that means, the individuals which are the highest fitness values should survive better compared to the worst fitness best individual.

So, this is why the selection is followed there and all the whatever the selection strategy basically targets; that means, that so that we can follow the Darwins principle there. And as we have learned it in a selection strategy, so all selection strategy anyway more or less favours the selection of better individuals in a current population and from where the mating pool can be selected; that means, it will allow mating to those individuals they are basically the best individuals. So, assuming that from the best individuals another best offspring will be obtained.

Now, in this regard the any selection scheme rather can be I mean compared or their effectiveness can be compared in terms of two concepts. One is called the population diversity and then selection pressure. So, in order to compare the different selection scheme that we have learnt first we have to understand about what is the population diversity, and then selection pressure. Now, let us understand the first population diversity and then then we will discuss about selection pressure.

(Refer Slide Time: 23:10)



Now, population diversity means if the ranges of fitness values are within a wider range, then we say that the population has very wide diversity.

So, as an example in this figure, so the range of fitness values is shown in this direction. So, these basically shows the population diversity or in other words if one population is in one population is this one only then it has lesser population diversity than the entire one. So, population diversity indicates that how many different that how what is the wider range of the fitness value that the population currently having.

Now, population diversity implies more exploration; that means, we can get the more accurate result. For example, if we consider this is the one population right and if we ignore this one, then may be that GA will stuck at with these optima, so this is the optimum one.

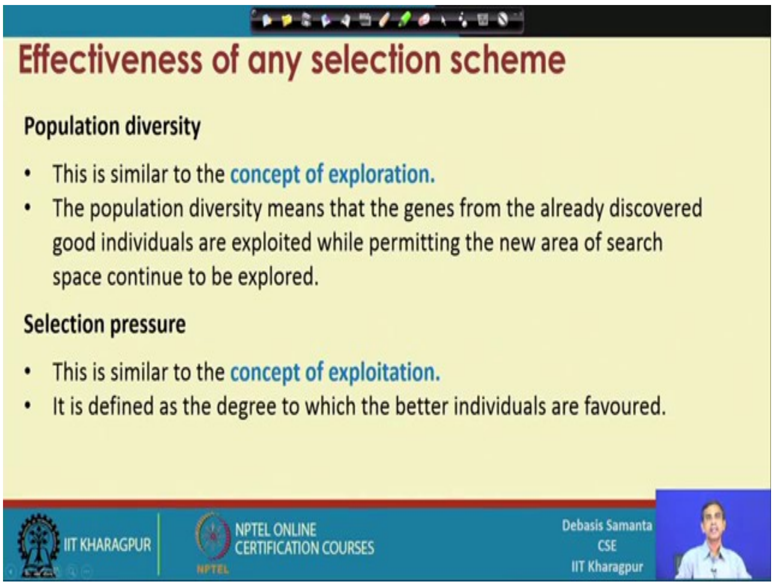
However, if you consider the population this one, then we may start at this one also. So, if we don't have the wider population diversity then we may trap into local optima. So, in other words, we can say the more population diversity means more exploration.

Now next is selection pressure. So, the selection pressure is basically, so it is basically what is the highest values that is there in in a current population. For example, if we consider, so this is the one population then selection pressure is this one this basically because highest fitness value this one. On the other hand, if we consider this is the population then this is this is the selection pressure, selection pressure is this one and this population has a selection then highest values of signal will this one.

So, here three different populations that we have consider having the three different selection pressures. So, here basically selection pressure means a lesser exploitation because whenever we get the selection pressure. For example, if we take this population and selection pressure is this one then our optimum value will be confined into these kind of things one.

So, basically will not be able to exploit the better results that can be had from the from the if the selection pressure is something else. So, this is the concept it is here. So, in summary, we can say that more population diversity means more exploitation and higher selection pressure means it is basically the lesser exploitation. So, it is basically exploration versus exploitation in the mechanism.

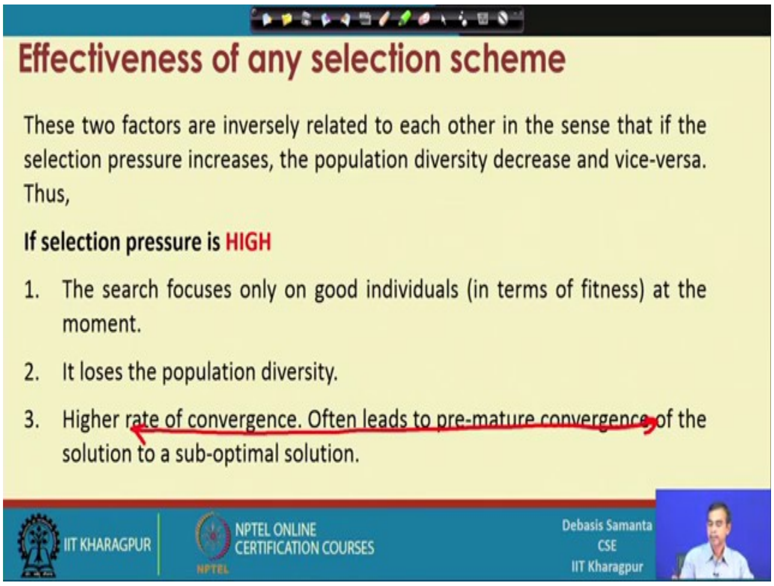
(Refer Slide Time: 26:00)



Now, let us see the how we can compare the different selection scheme anyway. So, population diversity as we have learnt this is similar to the concept of exploration as we have already mentioned, and then selection pressure is basically concept of exploitation, this means that it is defined as the degree to which better individuals are favoured.

On the other hand, so far the population diversity is concerned it means that genes from the already discovered good individuals are exploited while permitting the new area of search. So, basically the idea it is there if we follow the wider population then we have better searching capability, because we can search this one also we can search this one also and. So, many directions we can search it right. Anyway so these are the concept it is there.

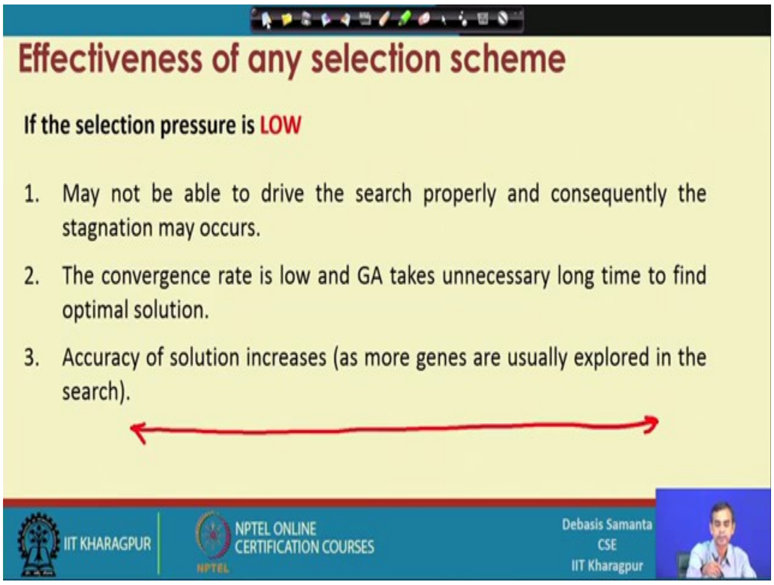
(Refer Slide Time: 27:00)



Now, we will quickly mention few points about if the selection pressure is high or low. So, first let see if the selection pressure is high then the search focus only on good individuals in terms of the fitness values at that moment and it therefore, loses the population diversity and then it basically leads to the higher rate of convergence that mean it will converge at the fastest rate and whenever the convergence is fast it may be premature convergence of the solution and therefore, the solution may not be, so accurate.

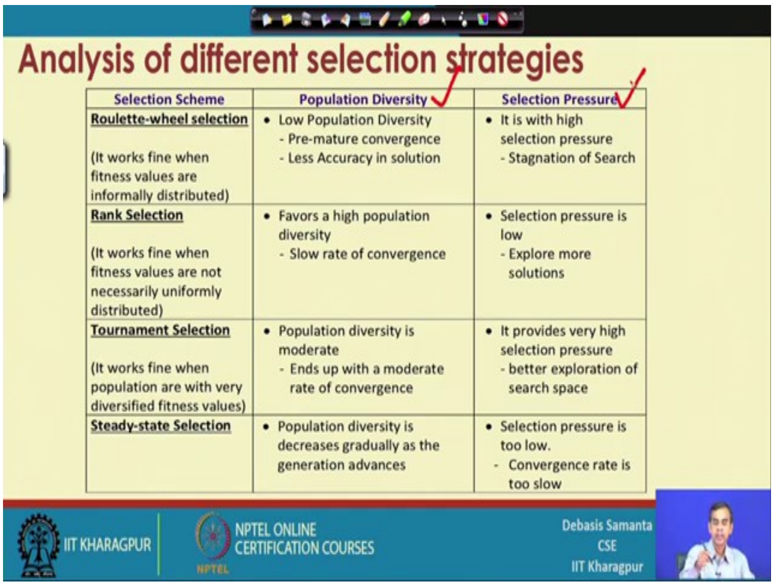
So, there is a basically trade off between the selection pressure and the population diversity here.

(Refer Slide Time: 27:51)



And on the other hand, if the selection pressure is low, so it may not be able to drive the search properly and consequently the stagnation may occur and the convergent rate compared to the selection pressure is high is low and GA takes unnecessary long time to find optimum solution, accuracy of solution. However, if the selection pressure is low increases as more number of individuals are explored in the search process.

(Refer Slide Time: 21:20)

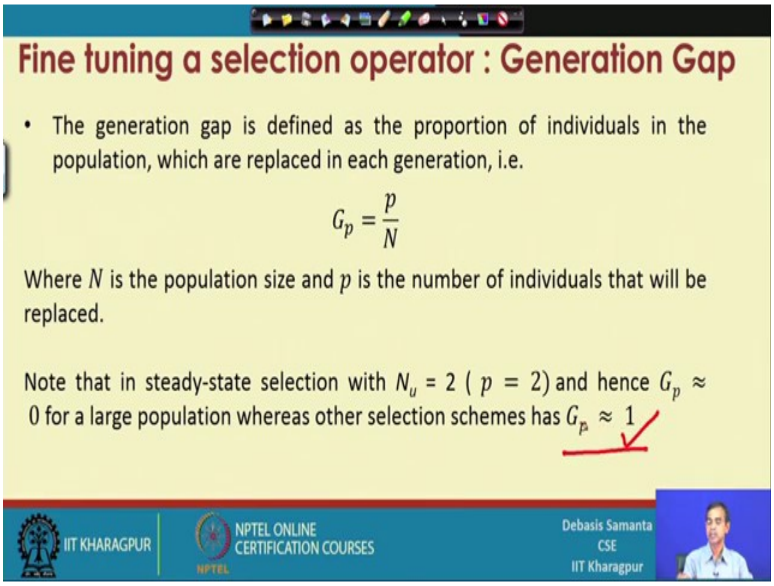


Now, I will just this is the last slide and I will quickly summarise the different what is called the techniques there in terms of population diversity and the selection pressure. Now as the table says that. So, for the roulette wheel selection is concern it provides low population diversity, whereas selection pressure is high.

On the other hand, in case of rank selection it favours high population diversity and then selection pressure is low. And if we consider tournament selection, popular population diversity is moderate and it provides very high selection pressure. Steady state selection on the other hand, population diversity is decreases gradually as the number of generation in advances where the selection pressure is too low.

So, this comparison gives us enough idea about which selection strategy are to be followed, if we want that high population diversity then definitely we can go for tournament selection. On the other hand, if we want very high selection pressure and then population diversity low then we can I mean choose for roulette wheel selection strategy. So, these are different selection strategy, that is there in the GA theory we have discussed in this lecture.

(Refer Slide Time: 29:39)

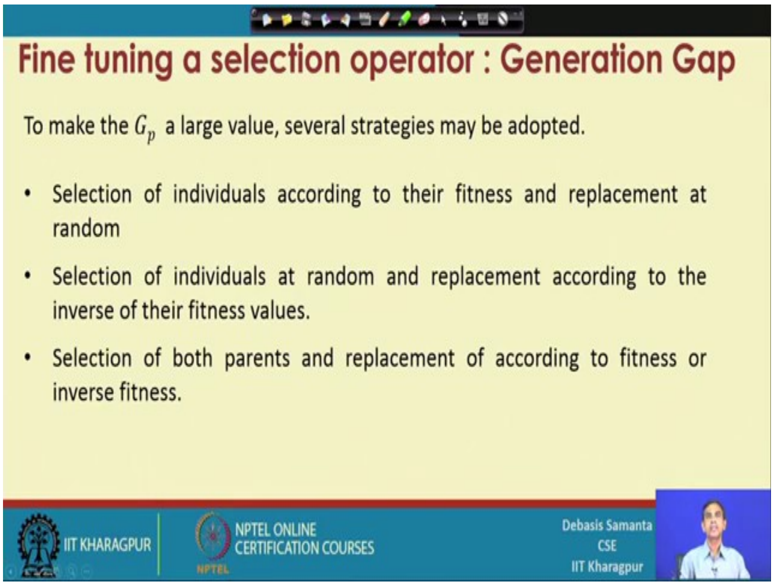


And we will discuss about other operators also there, it is basically called the generation gap, generation gap is denoted by $\frac{p}{N}$. Where p is basically the number of individuals those are basically will be replaced and N is basically the number of population is the size of the population.

Now, for example, in case of steady state selection, so G_p almost 0. That means, because if we take $n=2$ and for a large population size. On the other hand, for other selection strategy G_p equals to very high it is as high as 1. So, that the maximum value and the lowest value is 0.

So, G_p lies between 0 and 1, and G_p also can be considered as a measure about a selection strategy and it is usually preferable that selection strategy, which has better generation gap. So, in that case a steady state selection is not preferable because generation gap is very less ok.

(Refer Slide Time: 30:40)



We have considered the different selection strategies here, and our next topics that we are going to discuss is another operation, another GA operator. It is called crossover operation.

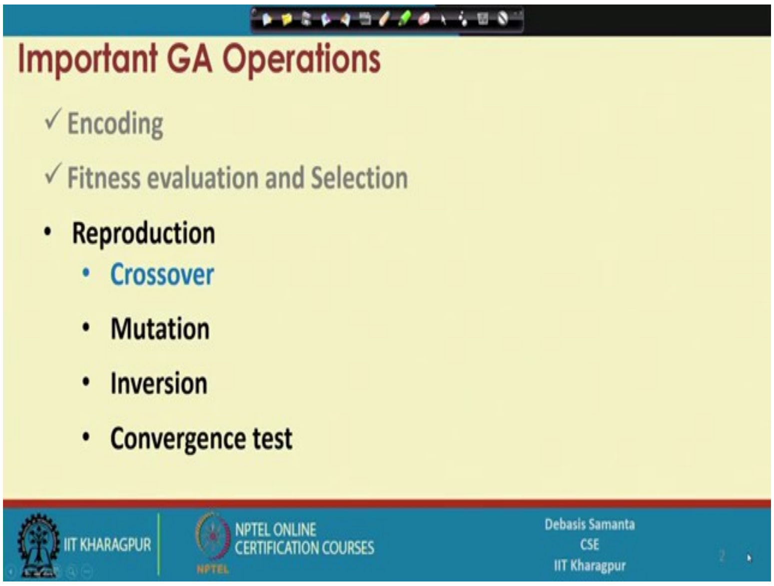
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 20
GA Operator : Crossover techniques

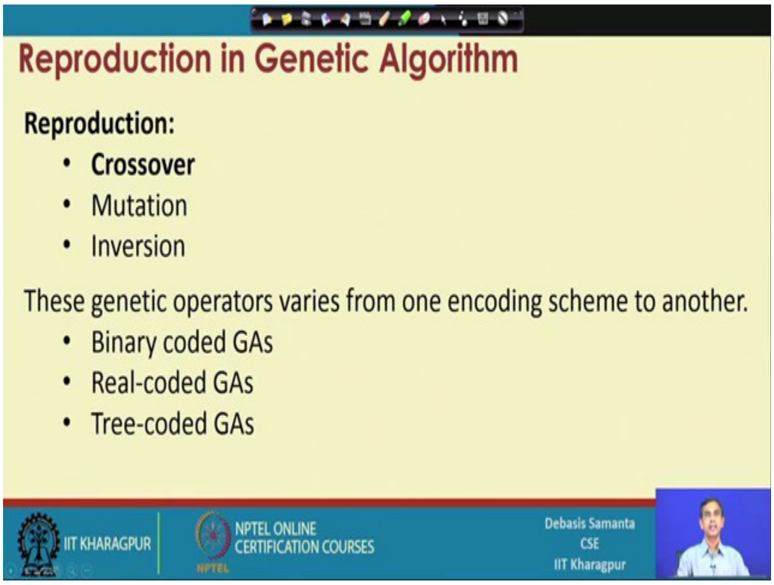
We are discussing, the genetic algorithm operators and we have discussed about encoding and then selection operations. Now we are in a position discuss another very important operation, it is called the crossover operation.

(Refer Slide Time: 00:33)



So, crossover operation is basically essential operation in the part of the another task in genetic algorithm is a reproduction.

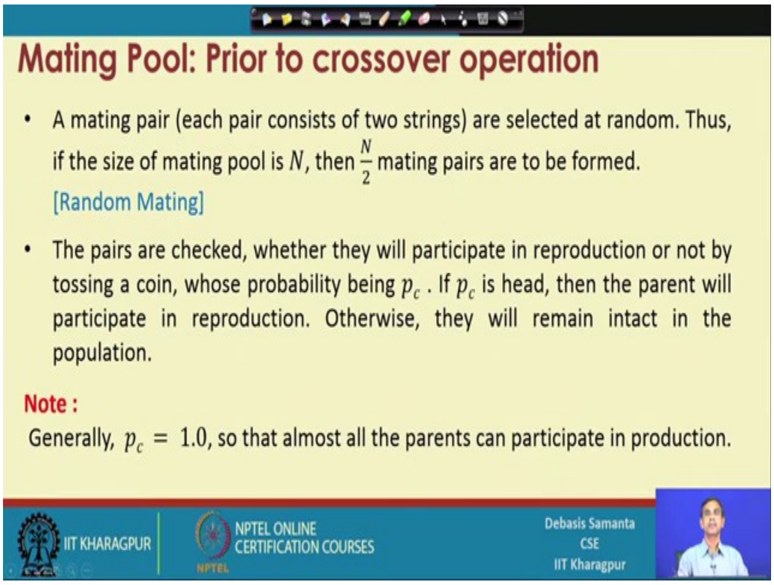
(Refer Slide Time: 00:45)



So, reproduction is basically, it consists of in addition the crossover there are some other operations involved mutation inversion which will be discuss in another lectures. So, first we will discuss about the crossover operation and we should, in this context I should mention that, the different encoding scheme. If we follow in genetic algorithm accordingly the different crossover strategies are to be followed.

For example, binary coded GA crossover is not applicable to real coded GA, or the crossover technique that is there for the real coded GA is not applicable to binary coded GA or tree coded GA. So, different the coding encoding scheme is.

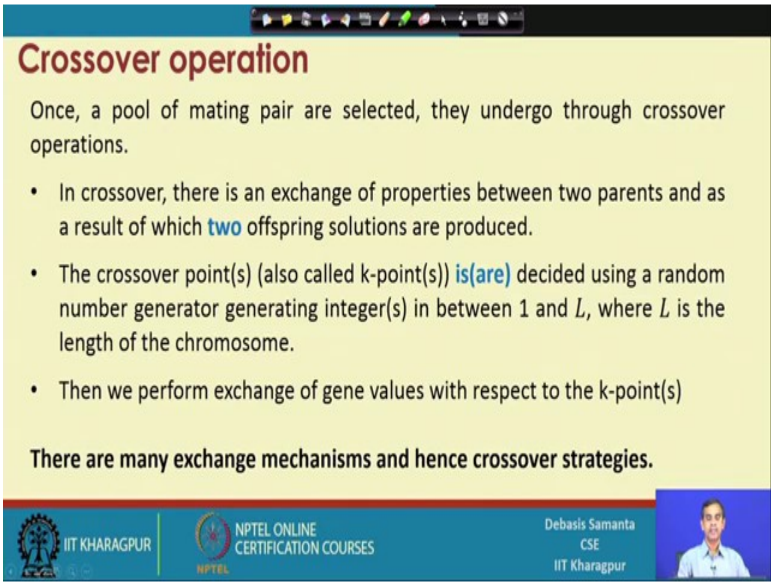
(Refer Slide Time: 01:29)



If it is there in the genetic algorithm and according the different crossover techniques are to be followed. Now, so we are discussing about crossover which is basically part of the reproduction and prior to this production the mating pools are to be created.

So, basic idea about the mating pool creation is that, if we select the N, p individual from there the mating pair are to be selected usually it is a random procedure; that means, two populations are to be selected at random and then they can be consider is one mating pair. So, this is called the random mating.

(Refer Slide Time: 02:08)

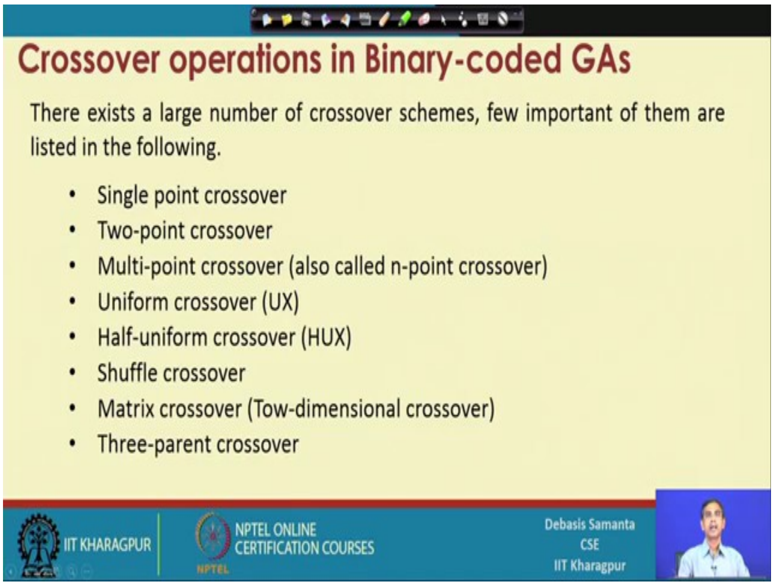


So, random mating once it is there, so it will produce the number of what is called the parents there. So, one mating pair this means two parents it is, and from the two parents we are to produce offspring. So, usually from a mating pair two offspring will be produced.

Now, what is the idea of the crossover mating is there. So, we know that each individual is represented by a string, that is whether is we are discussing about say binary coded GA of course, then it is binary string anyway. So, string is basically is the phenotype.

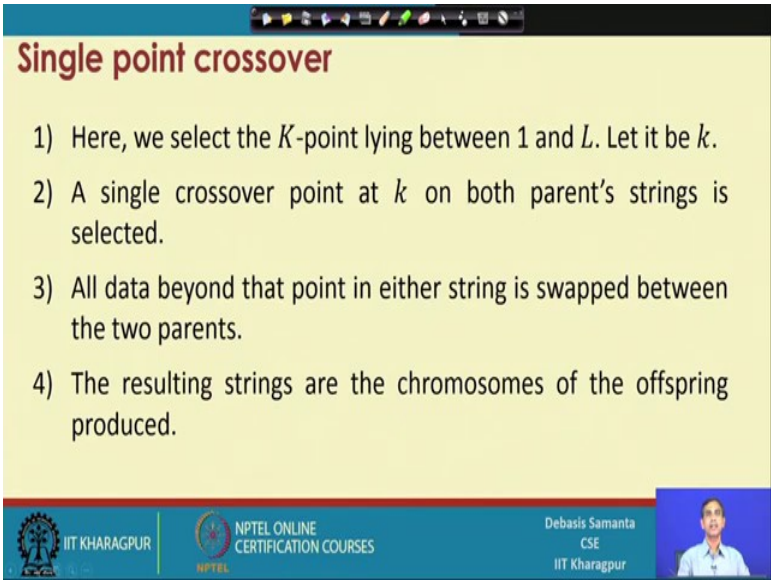
Now, for a phenotype, so there are different values are there and out of these values we have to select some points; that mean, it is called the k point. So, k point is basically as if it is the k point the kinetochore point in case of chromosomes and then based on this k points we have to take the crossover, crossover means is a mutual swapping or interchanging. So, this is the basic idea in the crossover techniques and now let us discuss about crossover techniques in binary coded GA.

(Refer Slide Time: 03:19)



And so for the different crossover techniques, in fact, in binary coded these are the number of crossover techniques are there. So, we have listed around ten crossover techniques we will quickly discuss one by one with example. So, the first is single point crossover technique.

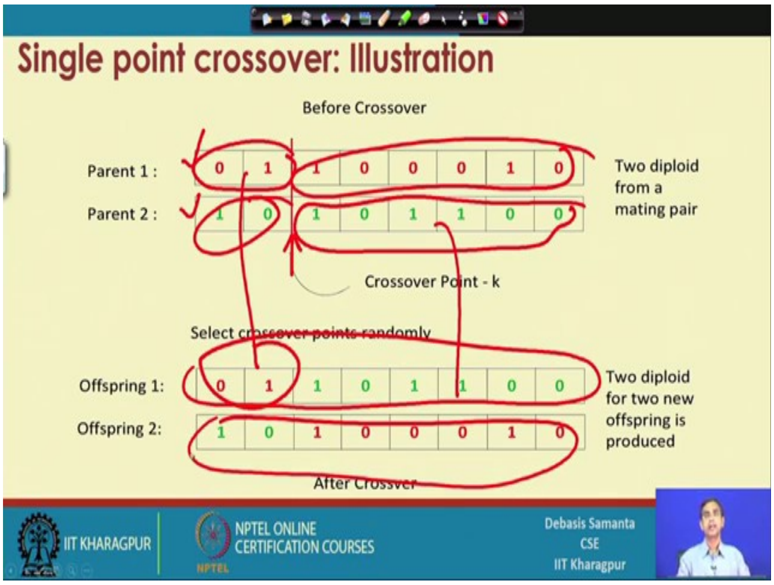
(Refer Slide Time: 03:37)



So, in this crossover technique, the idea is that if the length of the chromosomes is L, L means here the number of bits is L together our size of the chromosome is L, then we have to select one k point and let, it be the k where k is in between 1 and L. So, this basically decides where is the point k it is there on the chromosome, and this is a single point crossover because only one point is selected as a kinetochore point, so it is there.

Now, a single crossover point say at k is selected on both parents is there, and then a data beyond the points in either string is swapped between the two parents resulting the two offspring. So, this is the strategy, the strategy can be better understood if we explain with an example.

(Refer Slide Time: 04:28)

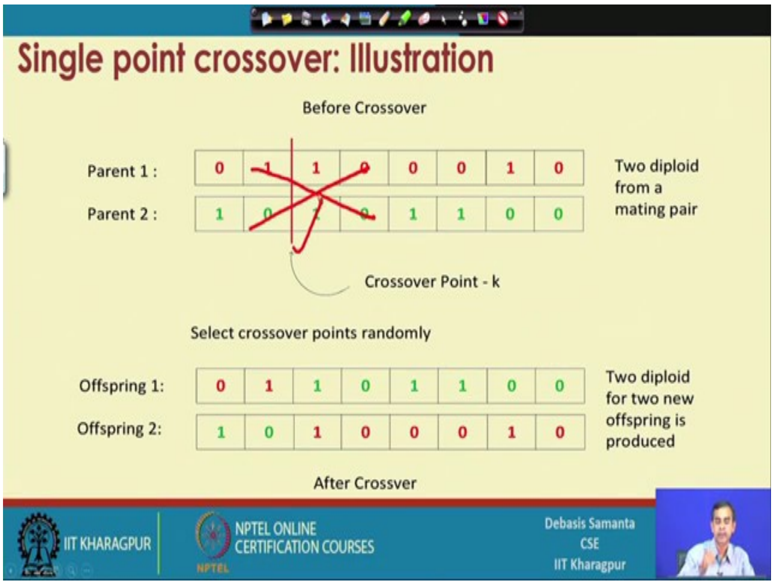


So, here is the example for this so suppose say this is the Parent 1, and this is the Parent 2, two parent’s chromosomes are there. And we randomly select one crossover point this is the randomly select crossover point k, and once it is selected this crossover point. What we have to do is that, we have to interchange the chromosome parts in their parents to produce the offspring.

For example, so this is the first part of the chromosome in one parent, and we take the second from the remaining part of the chromosome from the second parent, and if you take this one then it will produce another offspring. So, these part is from here and these part is from there.

Similarly, for the next offspring we take this part here, and then this part there. So, you produce the another offspring. So, this is the idea about the single point crossover technique, so idea it is like this.

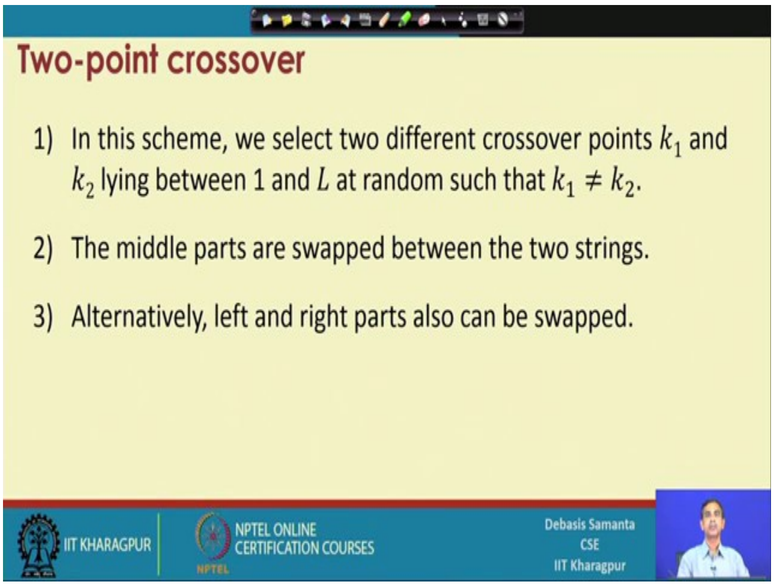
(Refer Slide Time: 05:30)



So, these basically these are the two crossover, and these are the two things are based on this kinetochore point it is there this one. So, the idea is like this and it is simple and also gives better result, but if this mechanism has this one limitation all those limitations will be discussed in due time. So, this is the concept of single point crossover technique.

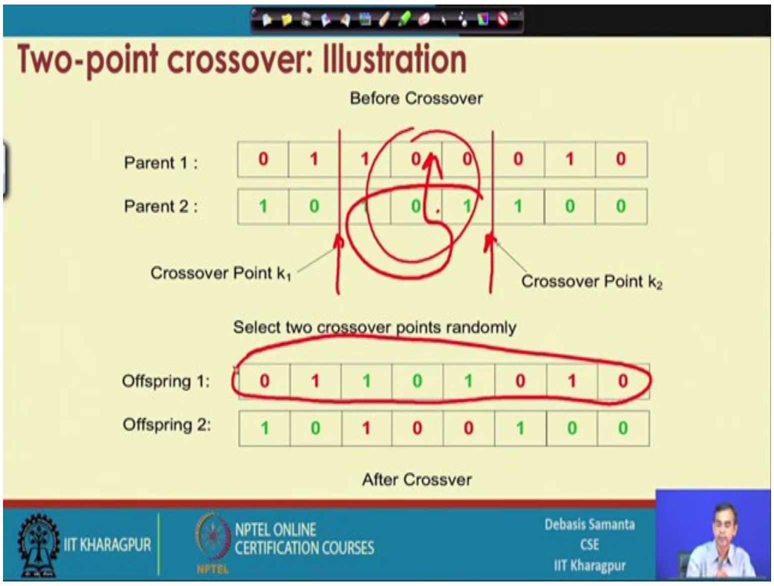
Next we will discuss about two-point crossover, it is just a modification or improvement over the single point crossover as the name implies instead of k only one point they are in the single point crossover.

(Refer Slide Time: 06:07)



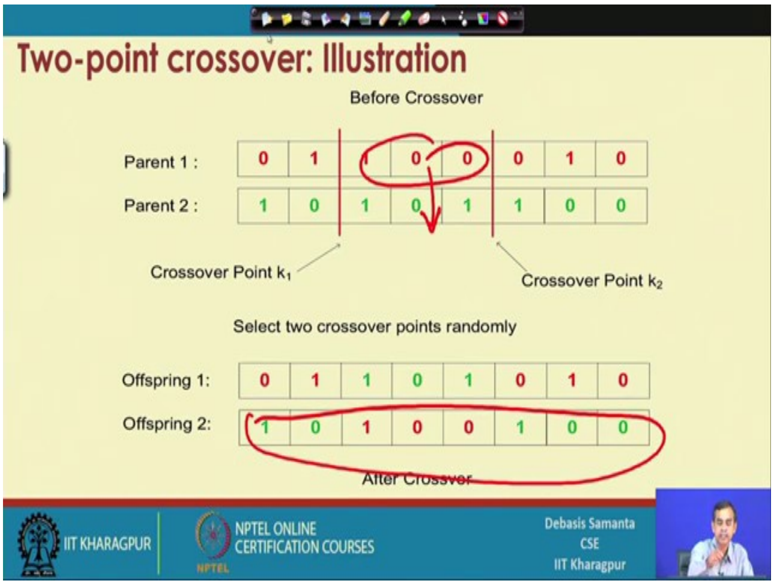
We have to decide the two points. So, let these two points be k_1 and k_2 in between the chromosome length L . Here, in this strategy the middle parts of the two parent's chromosomes are swapped to produce the offspring, so let us see an example.

(Refer Slide Time: 06:26)



Here is the illustration, so in this example we select one k point here, another k point here. So, it is say k_1 and k_2 , and then the offspring will be obtained by swapping the their middle parts so; that means, here this part is swapped between So, if we take swap means, if this part is go there then it produce this offspring and if we consider of these part is here right.

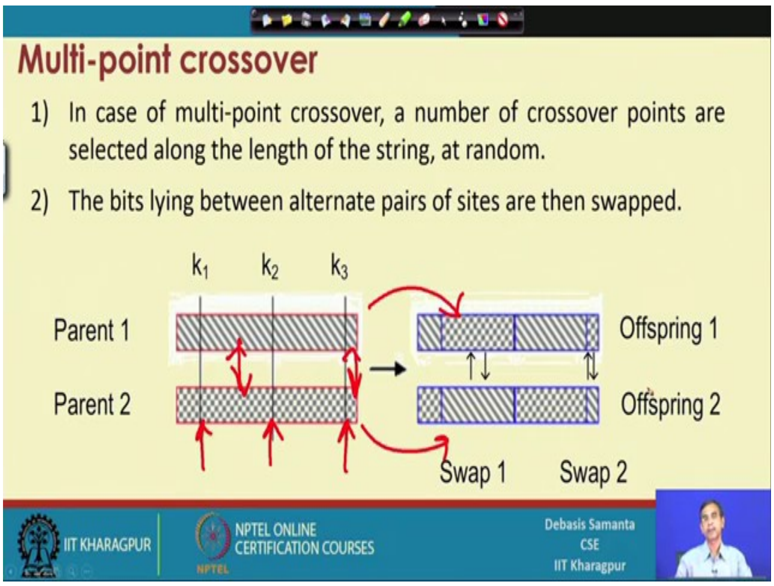
(Refer Slide Time: 06:58)



And then it will come this offspring, so this way the two offspring can be obtained, the two solutions. In fact, two new solutions are obtained, so this is the, this is why reproduction is carried out in genetic algorithm.

Now, so this is the two-point crossover technique, and then we will discuss about more general crossover technique is called the multipoint crossover technique. It is more what is called the more points are multipoint means more than two points are considered here.

(Refer Slide Time: 07:31)

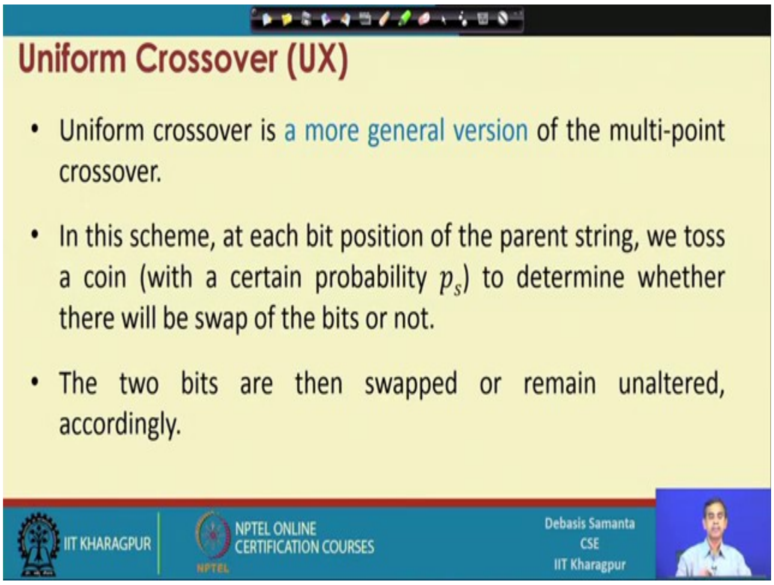


So, idea is like this so in this example we consider the multipoint as a three. So, we randomly decide three k points k_1 , k_2, and k_3 . And then producing the offspring is based on swapping the alternating part; that means, we can swap this part and this part, and then this part and this part.

So, this way another offspring will be obtained and here another offspring will be obtained which is shown here. So, it is alternative or alternatively we can also consider this part swapping and this part swapping there is another offspring will be also produce. So, it is the strategy whether odd number of parts will be swapped or even number of part will be swapped and accordingly the two different offspring will be created.

So, these are this is the basically generalisation. In fact, it is also two point if we consider number of k points only two, it is also same as single point if the number k point is this 1. So, we can say that multipoint crossover is the generalisation of single point as well as two-point crossover mechanism. Now, let us discuss the another crossover mechanism, which is which has its own benefit is called the uniform crossover mechanism, it is also abbreviated as UX.

(Refer Slide Time: 08:54)

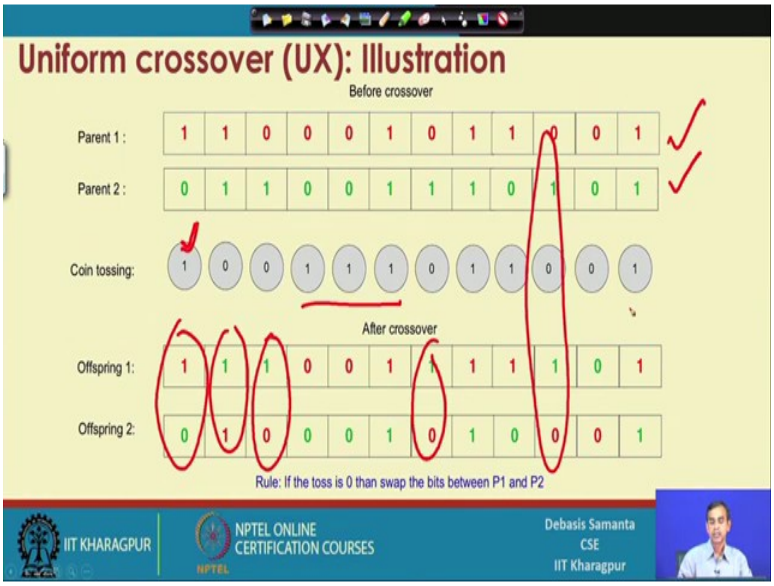


Now, in this mechanism, it is basically is a more general version of the multipoint crossover mechanism in fact. Now, in this process in this mechanism, we basically consider the number of k points same as the number of bits or number of bits, in the chromosome, in fact, that means, each bit position of the parent string we can consider k point, but here the thing is that we can make a toss for each bit position, whether they will decide or they will be swap or not.

So, swapping will result a new what is called the offspring. So, the idea it is that we have to consider for each bit position, and then take a toss, toss means whether 1 or 0, toss a coin like. And then with a it is basically assume certain probability, if it is a 50% probability that, in case of 50% you just half of the time you swap them.

And if the probability is 1; that means, here the p_s value that we have mentioned, here if p_s is the probability of swapping if it is 1, then you swap whatever the things it is there. Now, we can discuss this thing with an example, so that you can understand it.

(Refer Slide Time: 10:17)

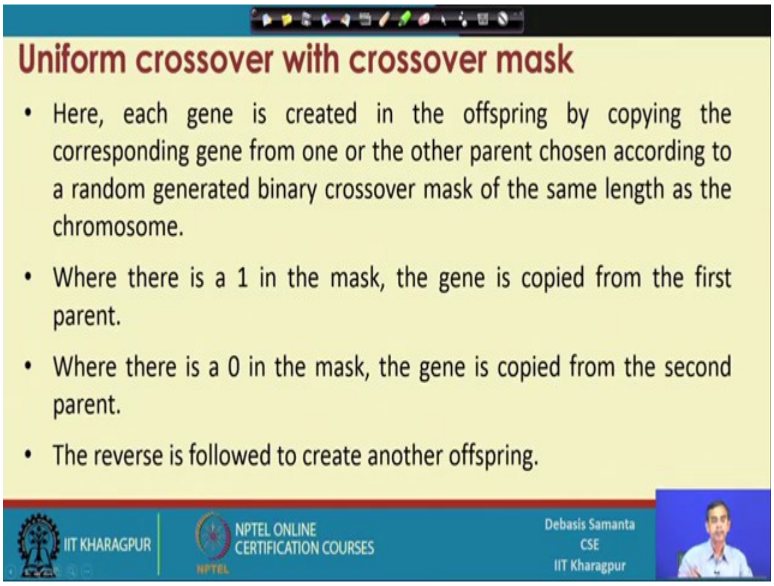


Now, here is the idea about let see these are the two parent chromosome p_1 and p_2 right and. So, for the first bit position we toss, a we toss a coin and the if the toss gives the result 1. So, here the possibly if the strategy is that, if toss if the coin tosses a 0, then we should swap between the parent p_1 and p_2 and if it is 1 then we should not swap.

For example, here in this case the toss is 1. So, in the then they will not be swap so they will remain same as it is there in the parents on the other hand it is 0. So, we swap it is, so it is this swap and it is 0 also we swap. So, it is swap means it gives the new values and then they are 1 1. So, no change 0 1 it is also swapped from 0 to 1 and then it is also 0 0, it is also swapped it is like this and. So, it is there, so this one, so this way you can see from this mechanism. If this is the input parents scheme and this operation is allowed on these two parents scheme, and then it provides two offspring as the different, what is called the chromosome pattern or different string.

So, this way two offspring can be obtained from the parent chromosomes using the uniform crossover technique. Next, we will discuss another technique it is called the half uniform crossover. So, the idea it is like this.

(Refer Slide Time: 11:52)



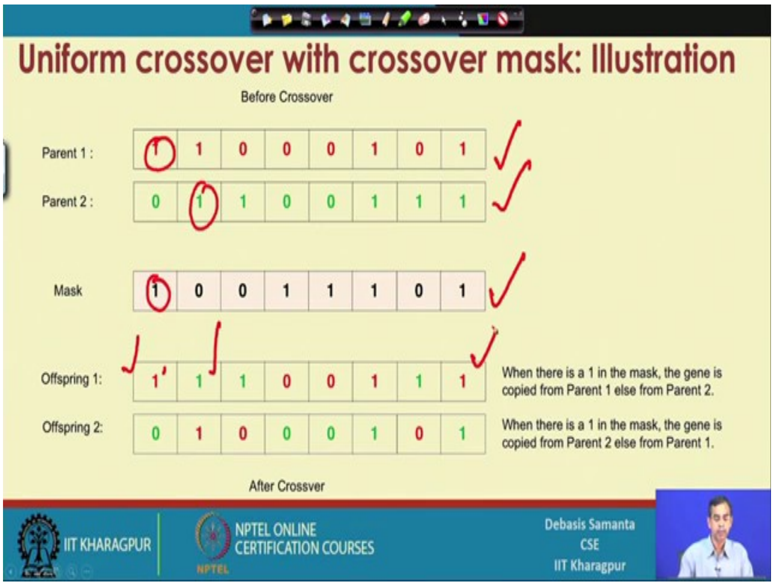
So, before going to half uniform crossover we want to discuss one little modification of the uniform crossover this is called the uniform crossover with crossover masks. So, there basically we have to toss a coin for every bit position, which is sometimes little bit computational expensive because tossing a coin require some computational effort.

Otherwise, we can consider one mask, this mask is a basically random what is called the bit patterns, sometimes one any third chromosome, third population, third individual can be consider also as a masks because it contents 1 and 0 in random pattern.

So, now whatever with the masks it is selected, now in the mask if there is a there is 1 then the gene is copied from the first parent. If there is a 0 in the mask then gene is copied from the second parent otherwise.

So, this way it will give one offspring and if you follow the reverse protocol then it will follow the another offspring Now, let us illustrate the concept with an example about this uniform crossover with crossover marks here.

(Refer Slide Time: 13:01)



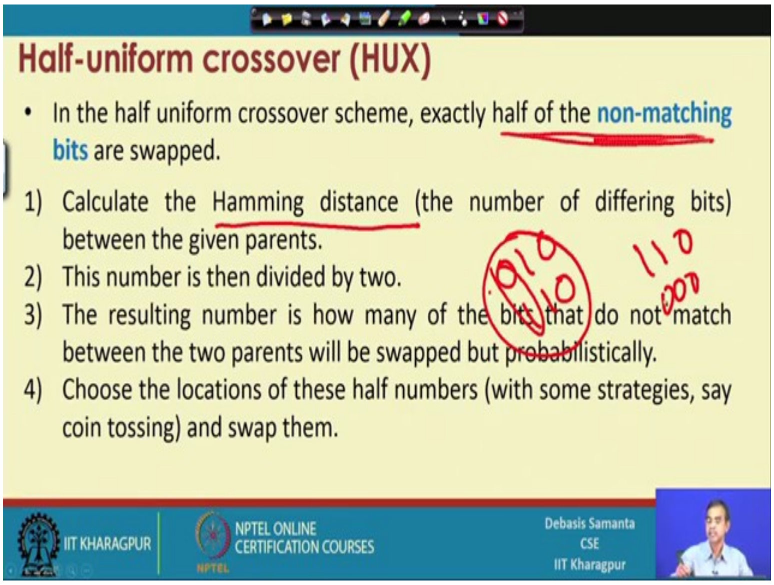
So, this is basically the two parents and this is the mask that we have followed. So, mask means it is a random bit pattern with 0s and 1s.

Now, the policy is that when there is one in the mask then the gene is copied from parent 1 parent 1 else from the parent 2. So, if this is the policy by which this offspring is created. So, here the mask is 1, so we will copy the bit pattern from parent 1 to this one, and if it is 0 then we will copy the bit pattern from this one. So, this way this pattern will be obtained

Now, if we will follow the reverse pattern; that means, when there is 1 in the mask the gene is copied from the parent 2 else from parent 1 then. So, this offspring will be created from the two parent, based on this crossover mask. Now if we consider the different mask from the same parent. In fact, we produce another offspring, so changing the mask also from the same parent can be used to generate more than one offspring, more than a large number of offspring from the same parent itself. So, this is the one advantage of this idea it is.

So, this is the uniform crossover with crossover marks. Now, we will discuss about other improve methodology, it is called the half uniform crossover sometimes it is called the HUX.

(Refer Slide Time: 14:28)



Now, in this half uniform crossover scheme, if the idea is exactly the half of the nonmatching bits are swapped now again here the two calculations are required. So, you have to first calculate half the half of the nonmatching bits. So, this is required a calculation here, but this calculation can be done very easily, if we use the concept of hamming distance.

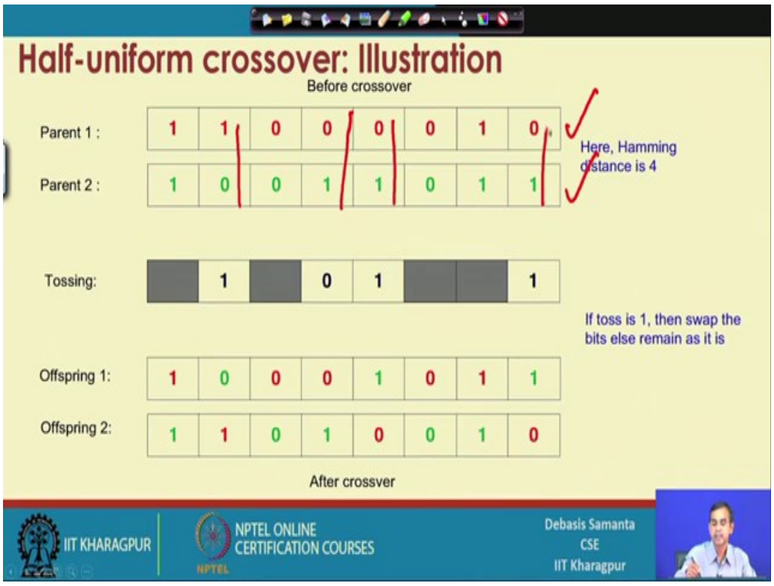
So, if the two bits things are given then the hamming distance is basically in the two bit patterns. how many bit patterns are different in each of them? So, it basically number of differing bit patterns gives the hamming distance.

For example, if this is 0 1 0 and 0 1, so it is 0. Then in this case we can see only bit pattern is different. So, hamming distance between the two is only 1, on the other hand 1 1 0 and 0 0 0 here we can see the two bit patterns are difference. So, hamming distance between these two strings are 2. So, this way the hamming distance gives us how many number of bits are differing in the two chromosomes.

Then half of them the number will be obtained just simply divided this hamming distance by 2. So, this one if the hamming distance is 6, and then the half of this will be 3 and if it is a 5, then it will be 2 like this one. So, these are the things it is there, then the idea is that whatever the half of the nonmatching bit patterns are to be swapped and they can be swapped, either in a random fashion by tossing a coin or just using any procedure here.

So, here the resulting number is how many of the bits that do not match between the two parents and they will be swapped probabilistically, here is a concept is called the probabilistically; that means, again we can toss a coin like. So, you can causes toss a coin and then if the toss gives 1 then, this bit pattern will be swapped and if it is not then 0. So, probability in this case 50% then so it is half of the things will be swap like. So, now let us see let us explain this half uniform crossover with an example.

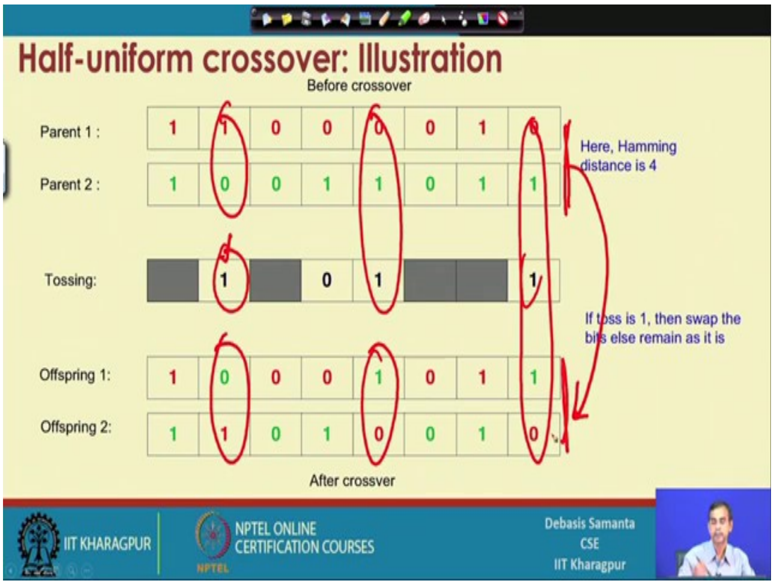
(Refer Slide Time: 16:42)



And here I just mention the example here. So, these are the two input chromosomes at the parents and we see in these two parents, the hamming distance is 4, here is basically this is the one different. So, different bit pattern is this one and this is the different, this is different and this is different. So, four bits patterns are different therefore, hamming distance is four.

Now, so for the different bit pattern we have to have a toss. So, toss is here say 1, so if it is 1 then the swap the bit pattern else it remains 0.

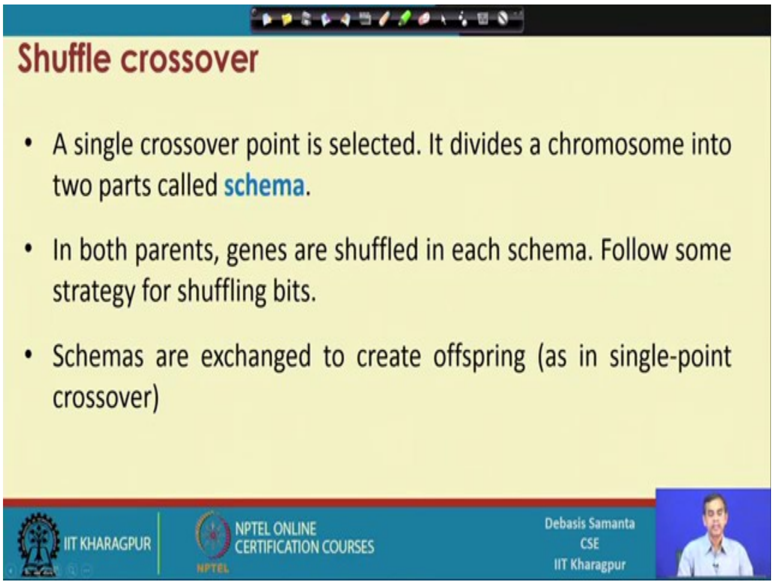
(Refer Slide Time: 17:27)



So, if we swap it then we can obtain, so here 1 and then swapping these two gives this 1 similarly here 1. So, it is swapping these two it will give this 1, and here is also 1 swapping these two it will give 1. So, this way from these two parents we can get two offspring.

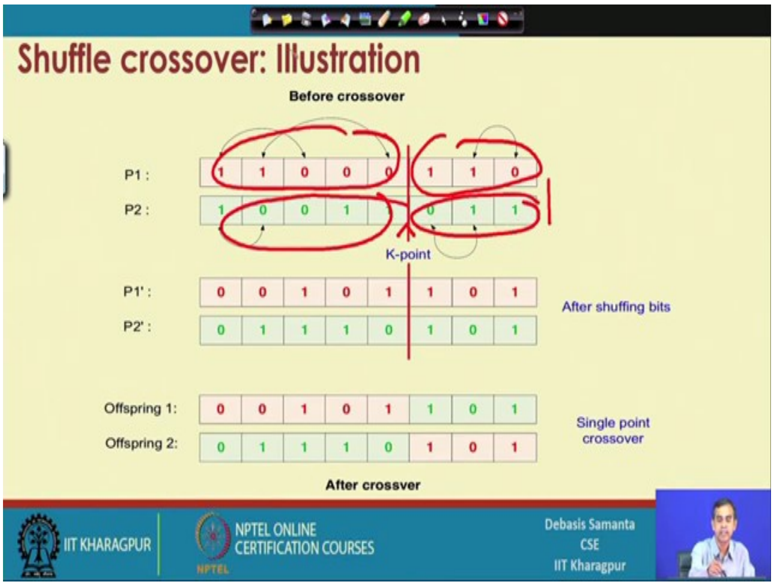
Now, again if we follow that different technique here for example, if toss is 0 then swap then another two different other two different offspring will be created. So, whatever the strategy whether toss 1 or 0 you can follow anyone, and then obtain the new offspring this way. So, this is the idea about the half uniform crossover techniques. Now, we will discuss a now another little bit complex crossover technique it is called the shuffle crossover technique.

(Refer Slide Time: 18:19)



Now, it is basically a single crossover look like. So, a single crossover point is first selected in this technique and then it divides the entire chromosome into two parts, they are called the schema and in both parents genes are first shuffled in each schema following some shuffling strategy. Then then as it is in the single point crossover the schema are exchanged to create offspring. Now, so here better we can explain this concept with an example.

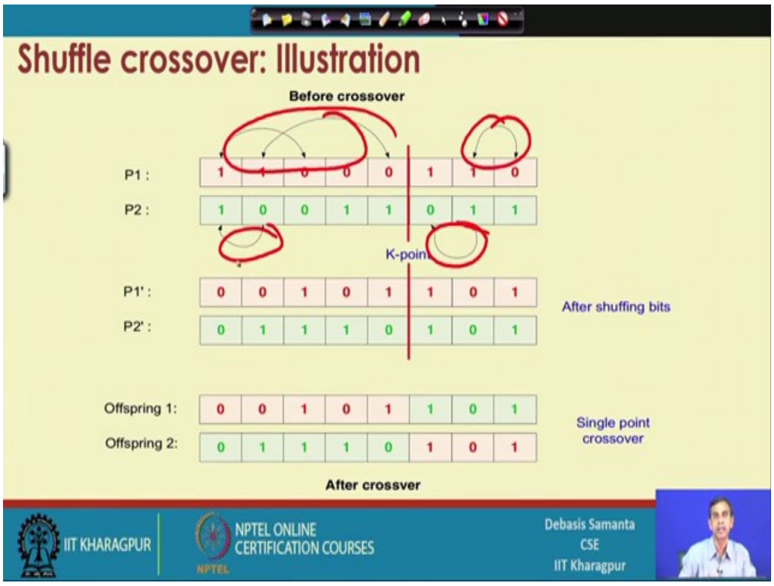
(Refer Slide Time: 18:50)



Now, let us observer this example here, so these are the two parents P_1 and a single point is selected let this single point is this one the k point and then this is the one schema in one chromosome and this is the another schema in one chromosome. So, the two schema in two chromosomes are selected in this case.

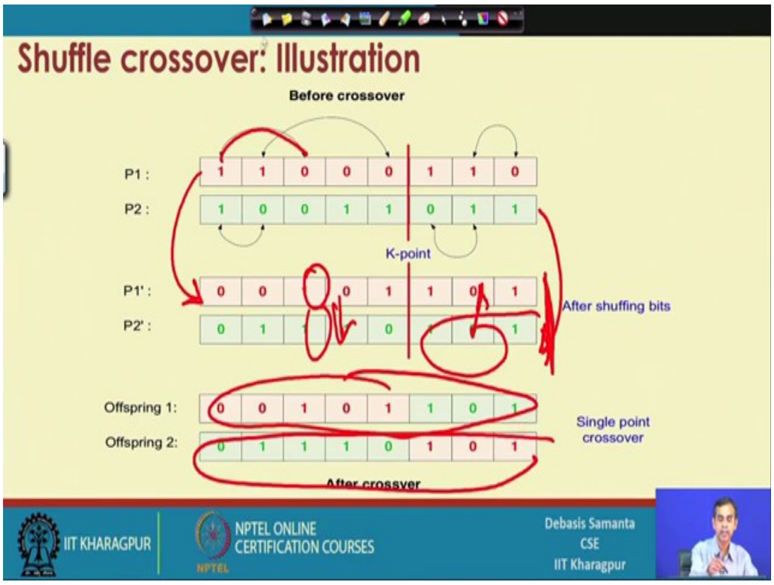
Now, in each schema we have to follow a shuffling, following some strategy maybe that first and third will be shuffled, and then second and fifth will be shuffled, in the first schema then here the last and first then next one will be selected and then first. So, these are shuffling, so these are the basically one shuffling mechanisms that we have followed.

(Refer Slide Time: 19:36)



And sometimes a may random suffling can also be considered. So, shuffling mechanism what will do so they will interchange this one? So, for example, from this shuffling mechanism, we obtain this one and for this this chromosome also we obtain this one.

(Refer Slide Time: 19:46)

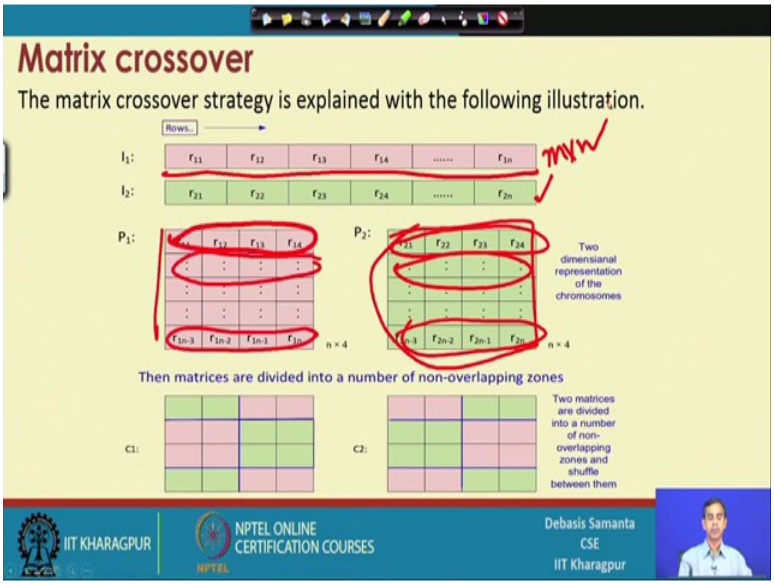


So, shuffling little bit results you the two different chromosomes in the parents. So, it is basically intermediate stage of the crossover mechanism. Once, it is then then we can follow the simple single point crossover thereafter, so it is basically if we follow the single point say. So, this part will be swapped this part, and then this part and this part will be swapped this one. So, it will give this one is the one chromosome and this another is the another chromosome.

So, finally, this is the parent chromosome and it will produce the offspring chromosome and we can understand that shuffling how it better compared to the or how this chromosome. So, far the variation in the offspring is concerned is comparable to the other crossover mechanism like say uniform crossover or the single point crossover like this one. So, this is a shuffling crossover it has little bit computational demand, but it is more variation in the offspring is possible.

So, if we note need more variation in the offspring then we should think for this kind of crossover mechanism, and then the another crossover mechanism it is called the matrix crossover, usually this crossover is preferable when the size of the chromosome is too large.

(Refer Slide Time: 21:17)



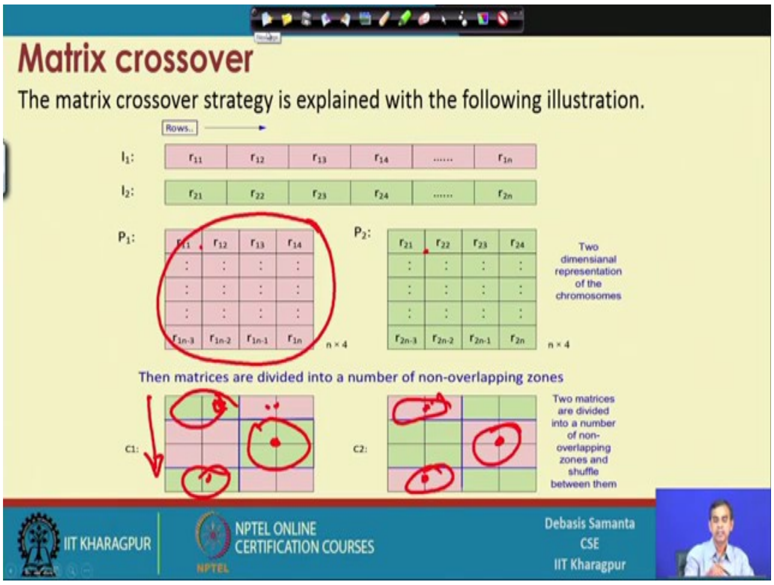
Now, if the size of the chromosome is too large then we can represent one chromosome in the form of a matrix. So, how a chromosome can be represented in the form of matrix is shown here, say suppose this is the entire what is called the chromosome for a parent and then we can make into an equal number of pieces.

So, for example, here four genes into one piece and then next four into one gene and then they can be placed in the metrics like. So, the first four in the first row, then second four genes in the second row and then this way so this one. So, it is called the row major ordering of the chromosome.

Similarly, for the second chromosome we can have another matrix in the row major positioning. So, this is the first row second row and the last row continues four elements also. So, it basically if we consider $m \times n$ like this one; that means, if in each row m number of genes and their n if the total size of the chromosome is $m \times n$ this way. So, a chromosome of size $m \times n$ can be converted into a matrix of m rows and n column that we have discussed here.

Now, one representing a linear chromosome into the form of a 2-dimensional representation of the matrix then we can follow the matrix crossover, it is just like a swapping crossover like or it is just like a shuffling what is the idea is that we have to make few block in each chromosomes. So, I consider that this is the one block, this is the another block.

(Refer Slide Time: 23:00)



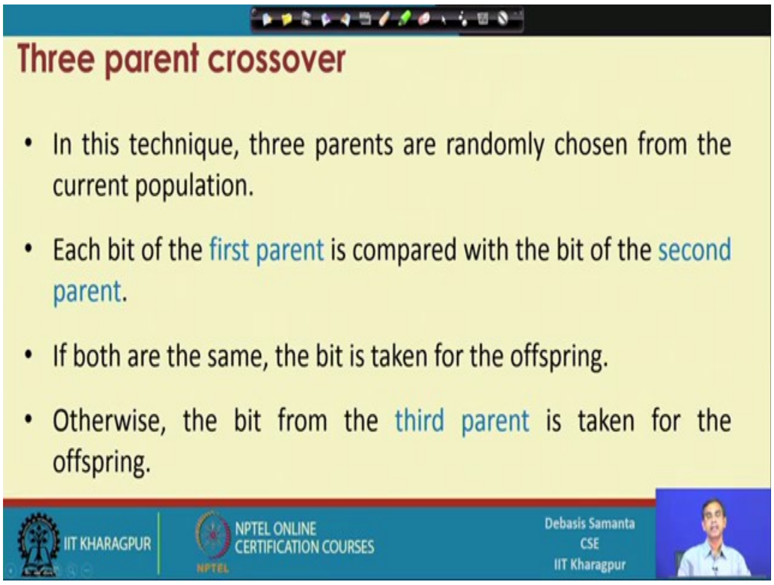
And this is the another block this is another block this is another block this is another block so; that means, entire chromosome is divided by six different blocks the same blocking is also followed here then we will swapped. So, this block will be swapped with this block, so here from this it will go there from this it will go there.

Similarly, this is the another block it will be swapped right from in between there. So, for example, it is not swapped here, so we swapped only this block is swapped with this one and this block is swapped with this one and this block is swapped with this one right.

So, if we can swap alternatively some blocks and then it basically gives, so this is the original what is called the chromosome and here is the this is the child chromosome or offspring chromosome, and you can say that in the child chromosome some parts these are the parts from the another parent similarly in this chromosome also these are the parts from the another parent and this is from the this one.

So, this way the variations or the new chromosomes can be obtained using this matrix crossover mechanism. So, we have discussed about the different crossover technique and there is a one another crossover technique it is called three parent crossover technique, the three parent crossover means, whatever the crossover technique that we have discussed they consider only two parents, but here is the three parents and the crossover technique is like.

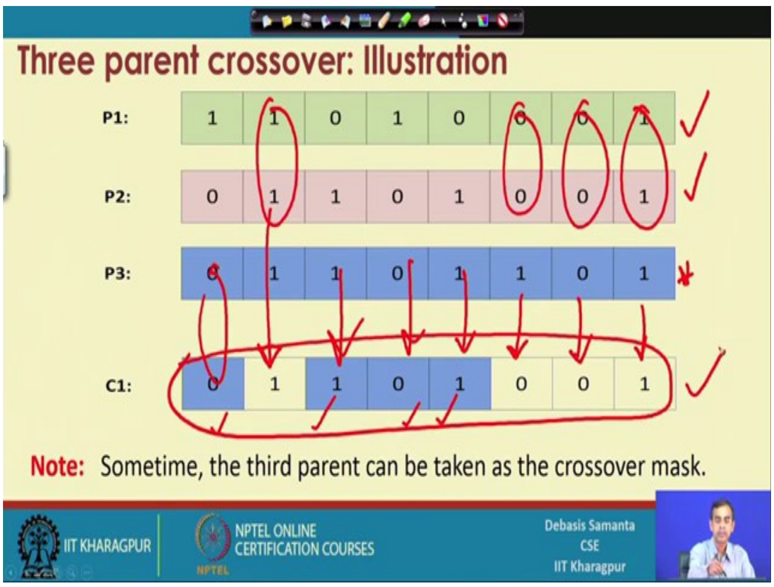
(Refer Slide Time: 24:38)



In this technique as you saw the three parents are chosen at random from the mating pool. Then each bit of the first parent is compared with the bit of the second parent, if both the bits are same then the bit is taken for the offspring otherwise the bit from the third parent is taken for the offspring.

So, this way one offspring will be obtain now again another offspring can be obtained if both are the same in the third step then we can take the any one from the offspring and this one. Now, let us see how this chromosome can be we can illustrate this kind of technique here.

(Refer Slide Time: 25:18)



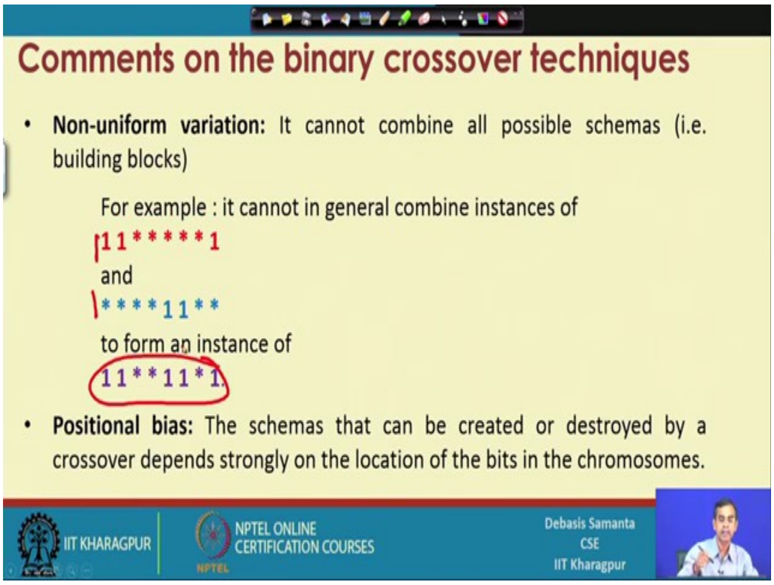
So, this is an example here, in this example you see this is the one parent and another parent is there, and then third parent if this one the third parent. So, as the strategy set if the bit position are same then we copy it into their for example, here the two bit position same, so it is copied into here.

And similarly we see here also the bit position are same. So, it is copied here and here also bit position same it is copied, here there also bit position are same it is copied here. So, we have copied directly from the two parents if we see that the two parents contain the same bit positions.

Now, for the remaining bits in the offspring we can select from the parent; that means, this is selected from this one, this is from here this is from here this is from here and so on. So, this way this new offspring will be created, so you can see we can see here that. So, three parents can be involved in to produce one offspring and it is a mechanism like this one.

So, this is the idea about the three parent crossover technique, and we have discussed about many crossover techniques, it is right time to discuss exactly who what are the.

(Refer Slide Time: 26:42)

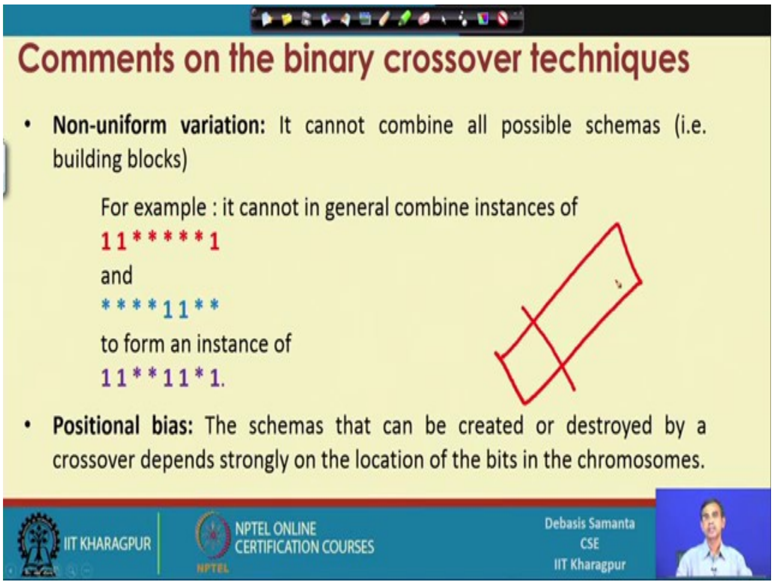


Why the different crossover techniques and what are the points are there to be considered in order to decide a better crossover technique, binary crossover techniques.

The first idea is that non-uniform variation is preferable non-uniform means it is not like that whatever the pattern there in the parent should be followed there. So, better if we can intermix the pattern there. So, for example, it is here if this is the one parent if the parent then, so it is basically better one instance will be like this one as a more variability is there.

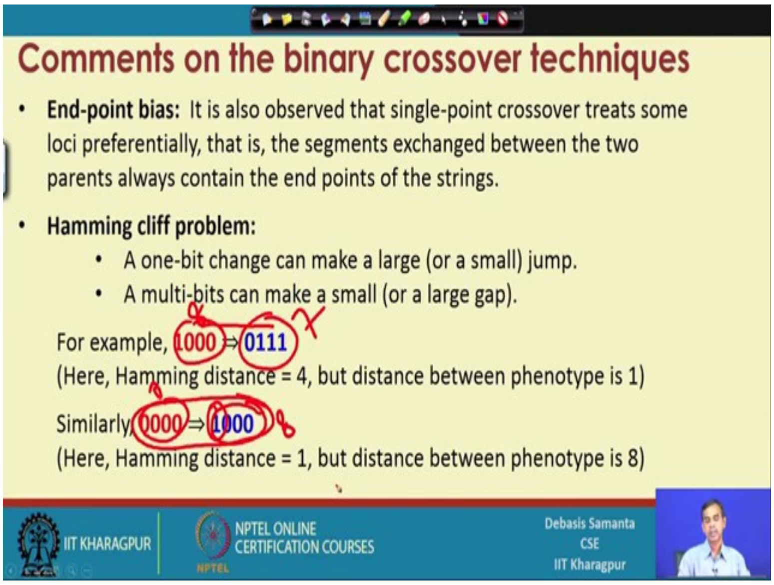
So, crossover technique should be such a technique, such that it should provide more variability in the offspring chromosome. Another is a positional bias, positional bias means.

(Refer Slide Time: 27:35)



If this is the one chromosome, so sometimes whatever the mechanism we follow either their n point will cannot be interfered; that means, n point will be copied always there. So, it is called the positional bias n point positional bias like. So, this scheme, so the crossover techniques should be that the positional bias can be avoided. Now, there is another also called n point bias it is also once at a positional bias.

(Refer Slide Time: 27:59)



For example, single point crossover always has certain n point bias always, because it is always n point is to be copied into this one. Another is that some crossover technique may suffer from one problem and this is called the hamming cliff problem.

So, hamming cliff basically if we change a small, it may not give a better changes. On the other hand, if we change again a large, then it can give a very small change for example, here if we see. So, some chromosome is there parent chromosome and it is converted to this one. So, what you can say that there is a large number of change changes in bits. Now, large number of changes in bits whether gives a better population from the wide diversity.

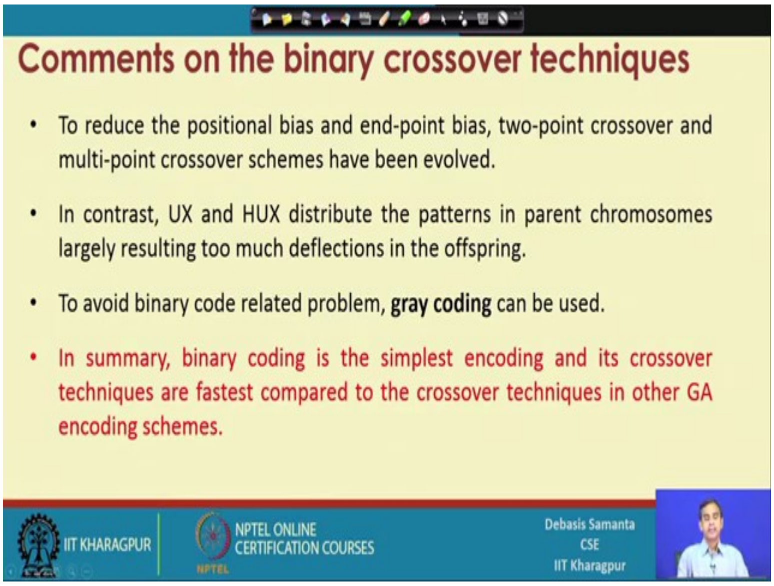
Now, for example here, so these are binary suppose binary and this is the equivalent is 8 and this is the 7, I mean decimal equivalent, this means that changing a lot number of patterns or bits basically produces the number changing from 8 to 7 is a very few differences there.

On the other hand, is another example here, if the 0 0 and there is a only small changes, this is the small changes. So, small changes can produce a huge difference for example, it has 0 and this is 8, so the one bit changes also can give a huge change.

Now, whenever we have to do this crossover basically our objective to explore the better what is called the solution better solution in the context that. So, we have a solution 1 and then the next solution should be near about that solution or very far from that solution it is need to be decided. So, if we see that one solution is far from the latest solution that we have explored then it may be good sometimes it may not be good.

So, hamming cliff problem is the one problem, which basically needs to be considered how much deviations that we want to have in our solution. So, these are the points that we have to consider. So, far the chromosome our crossover operation is concerned.

(Refer Slide Time: 30:22)



And there are many other ideas also, so we have discussed the different crossover techniques in the context of binary coded GA. So, sometimes other than binary coding people follow gray coding, it is the another concept which call basically also helpful to solve many of the problem that we have discussed; that means, position bias, n point bias or hamming cliff problem all these one.

Now, this is the last thing that we want to mention in the context of binary coded GA crossover technique. So, binary coding is comparable to any other coding technique. In fact, because whatever the crossover technique, crossover is the most what is called the costliest operation in case of GA algorithm

So, we should consider that crossover techniques should be in such a way that it is gives the faster result, because GA algorithm run execution takes maximum time in the process of reproduction; that means, a crossover operation. So, whatever the techniques are there it is observed that the binary coded GA, which follows the binary crossover techniques are the fastest technique.

So, that is why the many programmer they follow the GA encoding scheme, for the GA encoding scheme the binary encoding scheme and then implementation or programming of the different crossover operation also straightforward ok.

So, this is how much I want to mention about the crossover techniques related to the binary coded GA, and as I told you the different crossover techniques are to be followed in case of different coding is followed. So, next we will discuss the crossover techniques related to real coded GA.

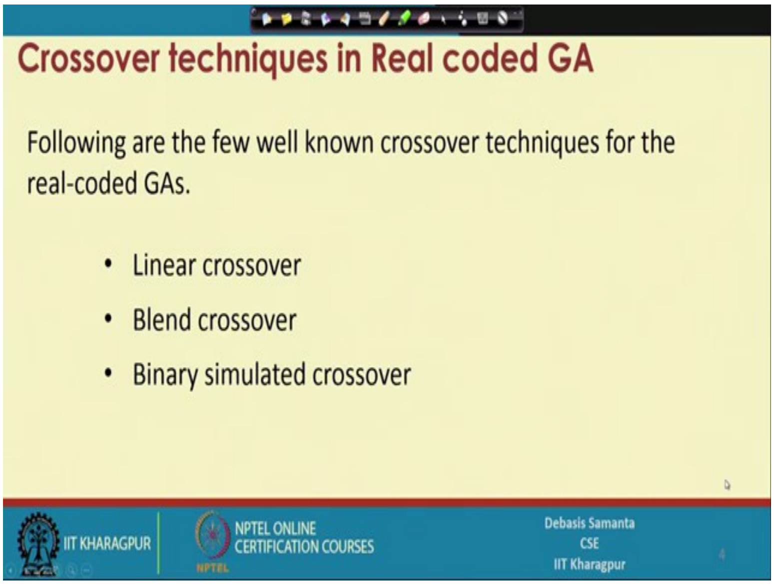
Thank you for today.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Science, Kharagpur

Lecture – 21
GA Operator: Crossover (Contd.)

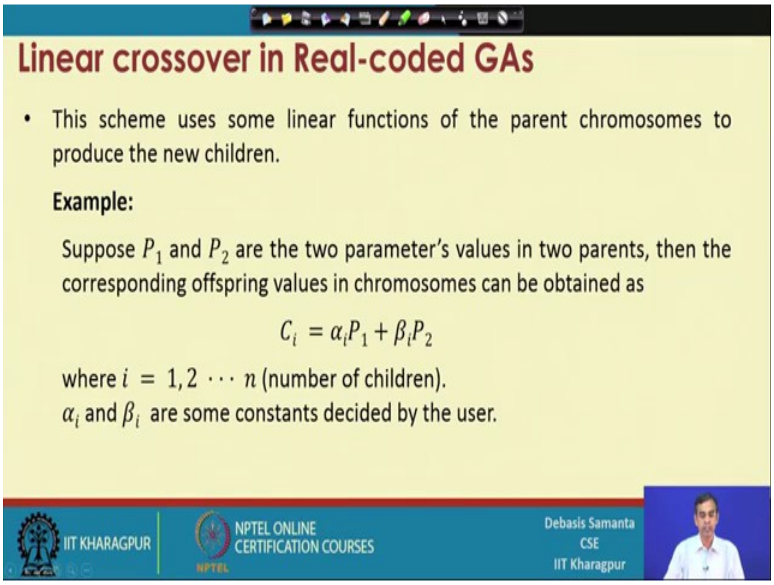
In this lecture, we shall try to learn the crossover technique; the crossover technique applicable to the real coded GA.

(Refer Slide Time: 00:31)



Now, there are three broad techniques which are followed to perform the crossover operation in case of real coded GAs. The three techniques are linear crossover, blend crossover, binary simulated crossover. So, these cross over techniques are based on the different policies in fact, so we learn the different policies that is here. Let us, first learn about linear cross over technique for the real coated GA.

(Refer Slide Time: 01:00)



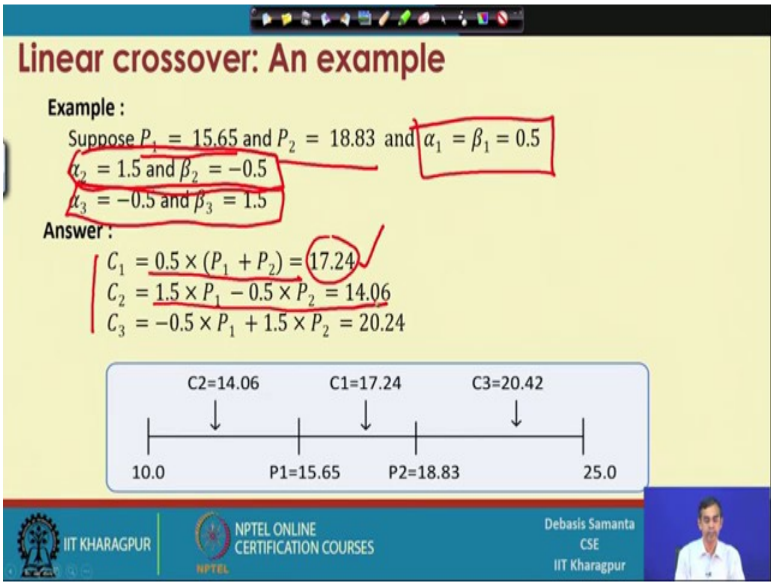
So, in this technique, we use some linear function that is why the name is called linear crossover; linear function of the parent chromosomes to produce the new children. So, we can discuss the technique better with an example; suppose, P_1 and P_2 are the two parameter values in two parents, basically they are the 2, any 2 genes values belong to two different parents P_1 and P_2. Then, the corresponding offspring values say it is C_i can be obtained using this formula. The formula says, that $C_i = \alpha_i P_1 + \beta_i P_2$. Here, P_1 is the gene values in parent P_1 and P_2 is the gene values for the parent P_2. So, it is like this, if this is the chromosome belongs to one parent P_1 and these are another chromosome belongs to another parent P_2.

So, for any gene value this one and this one, so it is basically P_1 and P_2. So, any two value gene values this one and this one, we denote this as a P_1 and we denote this as a P_2. So, having this structure, so that means, we want to calculate the gene values for the i^{th} chromosome, which belongs to the i^{th} children; that means, here one or more children can be produced. So, here we are considering how from the two parents P_1 and P_2 values the n number of children can be produced.

Now here, so the production is basically based on two parameters. These two parameters are called α and β. For the i^{th} children we will use α_i and β_i. So, these are the values α_i and β_i, if you want to produce n number of children will be decided by the user. So, it is give the responsibility decide the values of α and β s in order to produce the children.

So, this is the formula that is there, this formula is basically for how this i^{th} children's gene value can be calculated.

(Refer Slide Time: 03:29)



So, let us have an example. Say suppose, this is the one gene value belongs to the parent P_1 and another gene value belongs to parent P_2 is 18.83 and we are considering several cases or different values of α and βs. So, first case, let us take this is the value of α and β, in this case we take the two equal values which is 0.5 and then another case, where we take another α and β value and then third instance we can consider this one.

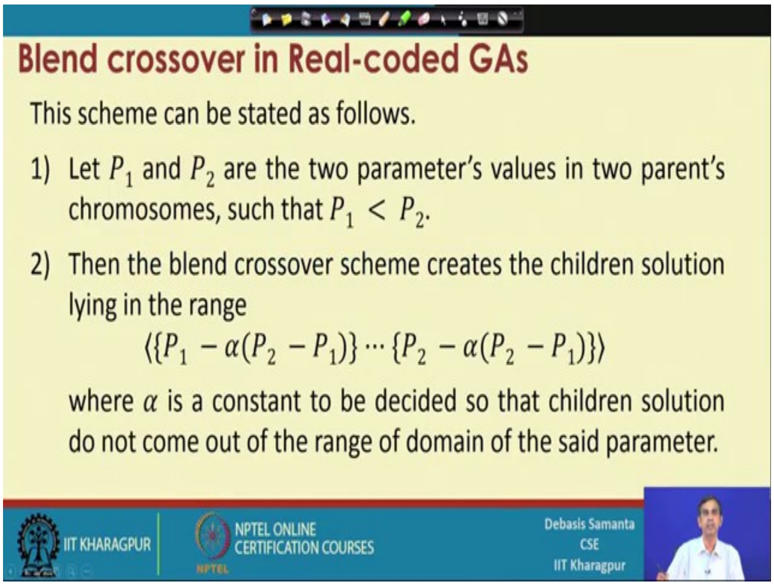
So, if we decide 3 set of values for α and βs, then we will be able to produce three children. So, how the three children can be produced in terms of this α, β values and given the gene values for the P_1 and P_2 is shown here. So, this is the first case, where the $\alpha \wedge \beta = 1$. So, it gives this is the gene value and for the 2nd case the $\alpha =¿$ 1.5 and $\beta =¿$ - 0.5. So, it gives these values and the 3rd case α is - 0.5 and β is 1.5, it will produce these value.

Now, so you have learned about how the linear function. So, these are the basically these are the linear function that is followed here. So, these are the linear functions and in terms of linear function we are able to calculate the children solution.

Now, here the different values of α, β has their own significance; have their own significances for example, here if we take α, β like this, then we will see that the children's value will be, so this is the parent value P_1 and this is a parent value P_2. So, if we take the α, β is like this, then the children will be within this. For example, in this case the children value is this one.

Now, if we take α, β like this one, α is heavy then β too, then the children will be beyond the P_1. Similarly, if we take this one α and β is this one, this case then it will this one. So, depending on the different values of α and β, so the children solution will be either here or here or here or within any regions, right within any point in the three regions can be considered. So, that is the importance of the different values of α and β. That means, if you have a good knowledge about the different what is called the values, a range of values, then you will be able to generate the different children's accordingly.

(Refer Slide Time: 06:34)

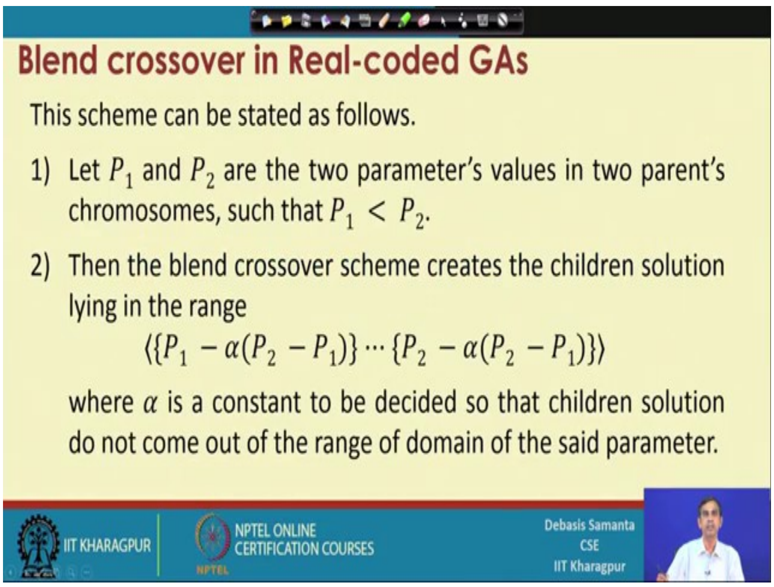


So, this is the idea about linear crossover, it is a very simple, but it has many advantage as well as limitation. The first is that, it is very simple to calculate because it uses a linear function and calculation of a linear function is straight forward and that is why this technique is very fast so for the computation is concerned and as we change the different values of α and βs, we will be able to generate large set of offspring from only two parents. So, this is the one advantage.

So, we can generate as many as solutions from the two parents and so this basically results population exploration and then, controls are possible to choose a wide range of change variations as I said in the last example, some values within the values P_1 and P_2, some values beyond P_2, some values beyond P_1. So, all these things can be possible, if we choose the values of α, β properly.

Now, here these are the advantage of course, the simplicity is the most important advantage in this case; however, it has certain limitation as well. The first limitation is basically the programmer should decide the values of α, β s and that is really headache for the programmer and in case of the programmer experience, then deciding right values for alpha and beta is really tedious job for the inexperienced user and more serious limitation is that if you do not choose the values properly, then the solution that will produce may leads to either premature convergence or stuck into a local optimum. So, there is a chance that this solution may not I mean with this cross over technique, the solution may not be optimum always. So, this is the advantage and limitation for the linear crossover technique. Now, another technique basically tried to address all those limitations, this technique is called the blend crossover; blend crossover in real coded GA.

(Refer Slide Time: 08:32)

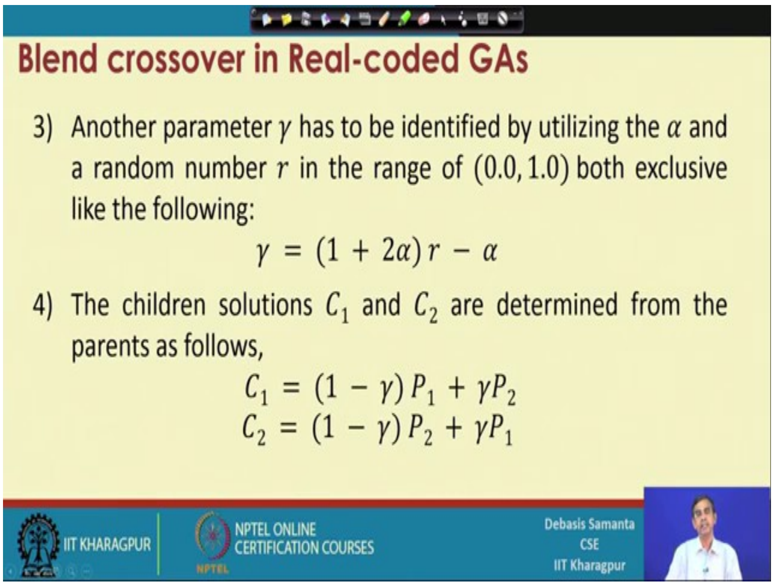


Now, we can explain this strategy like this, again will consider two gene values from parent P_1 and P_2, let they are represented as P_1 and P_2 and for simplicity will assume that $P_1 < P_2$.

Then, the blend crossover scheme it basically objective of the screen to create the children solution within the range, one is, this is the lower range and this is upper range within the range $P_2-\alpha$ (P_2-P_1¿ and these are lower range and this is the upper range $P_2-\alpha$ (P_2-P_1¿ . Here, the α value is basically decided by the programmer; that means how much you want to have the region wider or narrower, so that α value will decide.

So, it is basically is a constant and this constants should be decided by the programmer before using this operation. Once the α value is known to us, then we will be able to follow this technique.

(Refer Slide Time: 09:34)

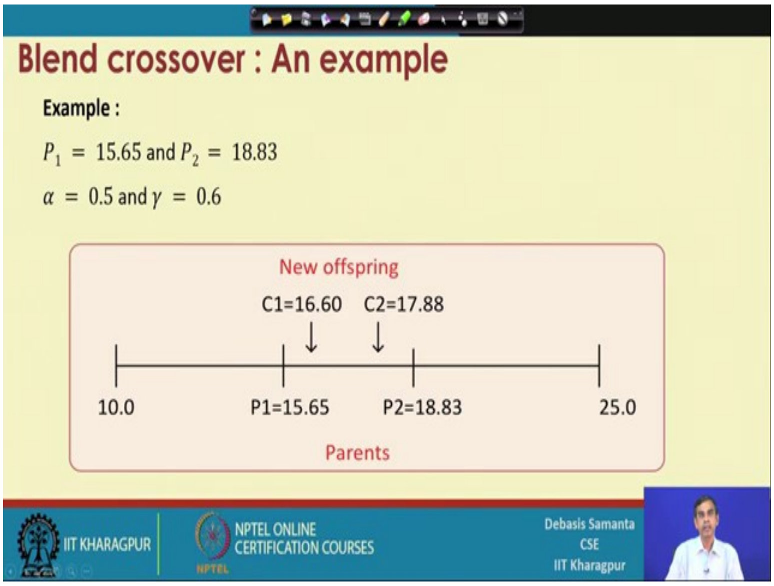


Now, this technique is basically calculates another parameters; now we denote this parameter as a γ and this parameter is denoted in terms of another random number r. So, α is already known, r a random number is generated a random, this random number r should be in the range 0.0 to 1.0 and then based on this r and α is already known to us, will be able to decide the value of gamma.

So, basically gamma is, in this case, a random value inside because r, as it is r is a random number and α is a constant, so gamma again it is a random number. Now, having this gamma random number, any two children C_1 and C_2 can be calculated taking the confidence of the gene values of P_1 and P_2, which is shown here. For example, so $(1-\gamma)P_1+\gamma P_2$ and the another solution is $(1-\gamma)P_2+\gamma P_1$. So, changing the values of P_1 and P_2 like this, we will be able to calculate this one.

Now here, the unlike in case of linear crossover technique, we can generate a large number of solutions C_1, C_2, C_3 if we take the different random values in fact. So, taking the different random values, we will be able to have the different γ values and the different values will produce the different solutions according to this blend crossover technique. So, this is the scheme in fact, in case of blend crossover technique.

(Refer Slide Time: 11:11)

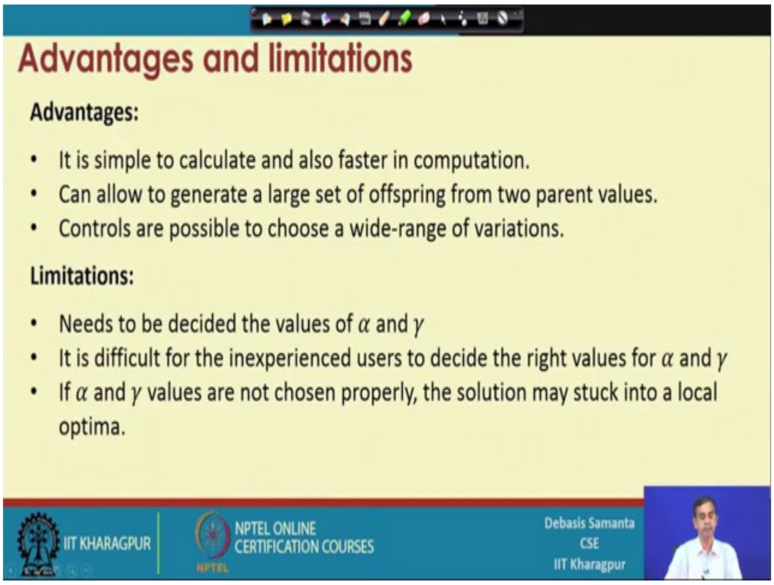


Now, let us illustrate this technique with an example, here is a simple example, let the gene value which is belongs to the parent P_1 is this one and another gene value belongs to parent P_2 is this one, we consider alpha is 0.5 and gamma based on some random number, which is not mentioned here say that gamma at the moment is obtained as 0.6.

Now, with this we will be able to calculate the C_1, that is $(1-\gamma)P_1+\gamma P_2$ - P_1 and this is basically the value that can be obtained for the one solution and another solution.

If we take some another random number, which will give an another γ, then another solution can be this sides or another solution can be this sides can be obtained. So, this way we will be able to generate a large number of solutions like a linear crossover, but only in terms of a probabilistic way random number.

(Refer Slide Time: 12:11)

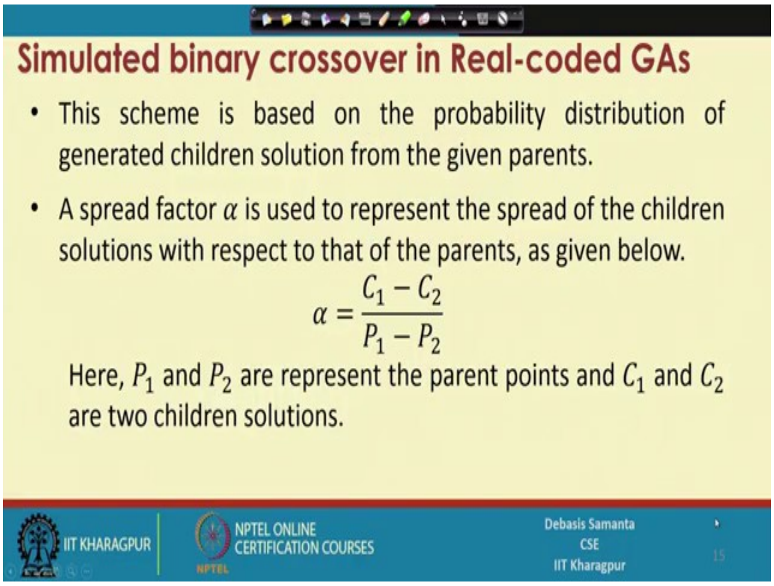


Now, so this is the plane cross over. Again, it has the limitation. It is also simple because it also follows linear one equation for the calculation and like linear crossover, it is also known to be a faster one technique and like linear crossover. It also produces a large set of offspring from any two parent values and controls are possible because here the wide range of variations can be possible, if we choose enough random number as we wish. So, this is the limitation. Like linear crossover technique, it is also the, is a good point of this technique is that it is simple and then fast.

However, it has the limitation, the first limitation is α, but again you can note that α can be chosen with certain calculation, that how much the range that you want to have? So, that α can be obtained little bit by a prior calculation and then so α calculation is not a big issue. Now, so α the calculation can be done by some estimation, then γ also can be done with the help of α values which are obtained and then just generating a random number.

So here, α and γ can be decided, but it can be decided little bit calculated manner as it is not possible in linear crossover. So, this is the one difference between linear crossover and blend crossover. Again, for the inexperience user, deciding α is little bit difficult. Although, it is not as such difficult as the linear crossover technique. So, it is little bit simpler for the user to decide the α value in fact and obviously, the α and γ are the two deciding values in order to decide the right chromosome values for the children. So, if we do not decide α and γ values properly, then it may lead to premature convergence as well as stuck to a local optimum solution. So, this is the blend crossover technique and we can understand that it has little bit it, it is comparable you compare to the linear crossover technique.

(Refer Slide Time: 14:37)



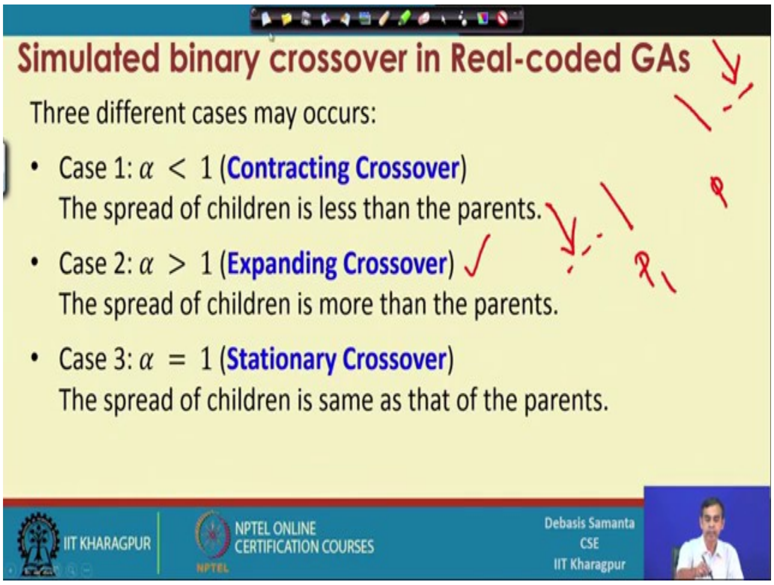
Now, we will discuss another technique. This is basically another statistical technique also called and it is called the binary simulated crossover technique and so in the binary crossover simulated techniques, the idea is more what is called a statistical in nature, is it gives I mean variation; more variation compared to the linear and blend crossover. However, this technique is little bit computationally expensive and you will discuss it about.

So, this scheme, the simulated binary crossover is based on the concept of probability distribution and then, they basically follow certain probability distribution function to generate the children solution, that is the one advantage and it has been observed that if we use the probability distribution function rather than the simple random number as we have discussed in blend crossover, it basically produce the better result and can avoid or can address the premature convergence and then start to the local optima. So, this technique is preferable in the sense that it gives better solution, then the solution that can be obtained following the linear crossover technique and blend crossover technique.

Now, the basic idea in this technique is basically, it considers one factor, it is called the spread factor. So, it is spread factor and denoted as alpha, the spread factor can be calculated by this formula. We have to assume any two C_1 and C_2, that means, how much variations in the children's solution that you want to have and these a P_1 and P_2 are the input value that you are having, then knowing or anticipating these are the C_1 and C_2 value, then you can calculate α. So, α calculation is basically under your control that how much division between the children solution that you want to have and decide, that will decide the values of alpha. So, α is a little bit can be calculated, so in that case user does not have to be an experienced user in fact.

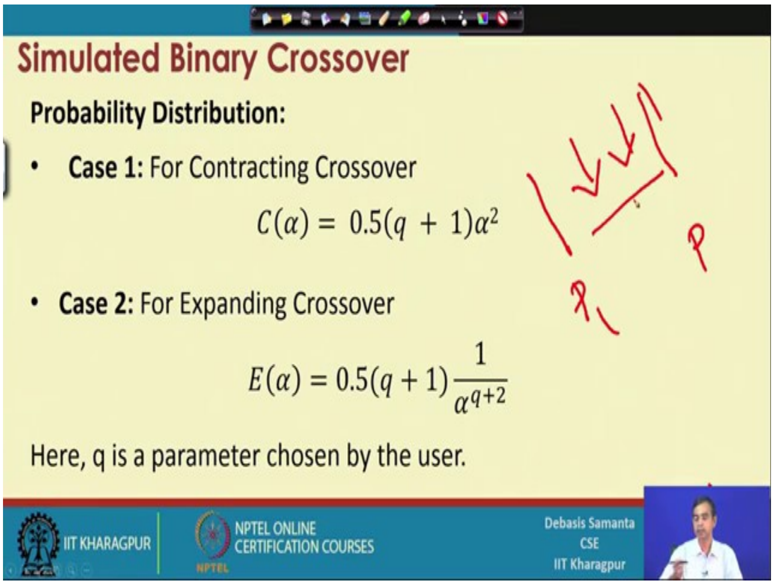
So, once the α, the spread factor is calculated, then we will be able to calculate the solution; children solutions which can be, which has the three different situation. So, will discuss about the three cases.

(Refer Slide Time: 16:39)



So, three different cases mean if $\alpha<1$, then the simulated binary crossover is called contracting crossover, so in fact, in case the spread of children is less than the parent.

(Refer Slide Time: 16:57)

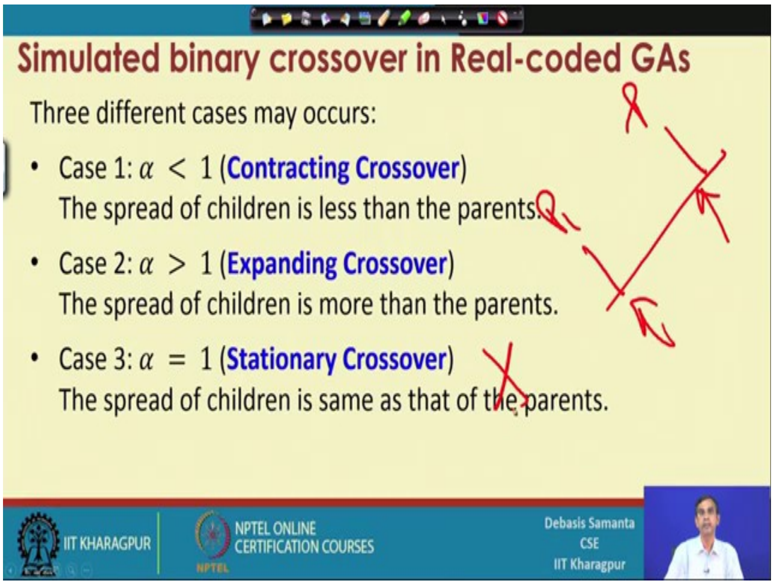


So, is basically if this is the parent P_1 and parent P_2, then C_1 and C_2 will be within this one. So, it is called the contract that means within the parent P_1 and P_2.

On the other hand, case 2, if, so case 1 is basically α < 1, it is called the contracting crossover. In this case, the P_1 and P_2 are the two parent values, then the C_1 and C_2 can be obtained anywhere in between the parent and P_1 and P_2.

Now, the second case is case 2, in this case $\alpha>1$ and in this case it is called expanding cross over. So, expanding crossover means if it is the P_1 and P_2 then C_1 will be calculated beyond P_1 and P_2 here or there.

(Refer Slide Time: 17:58)



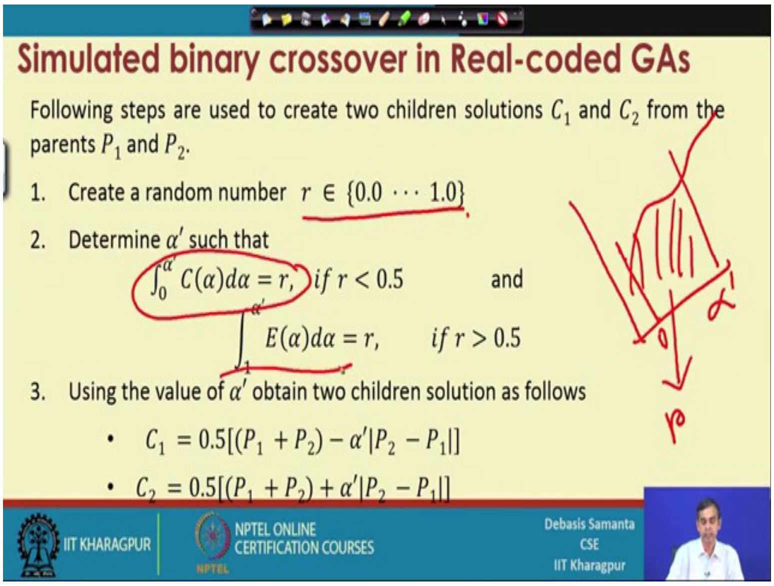
So, these are the expanding and the 3rd case is if $\alpha = 1$ and eventually $\alpha = 1$ means P_1 , P_2 and then C_1 and C_2 . So, this is not a useful fact because it will not produce any variation. So, usually this technique is not considered or the $\alpha = 1$ is not they acceptable value. So, will consider only this is alpha using these two cases that is $\alpha < 1$ than 1 and $\alpha > 1$, that is either contracting cross over or we have to use a simulated binary crossover as a expanding crossover.

Now, let us see, how the different crossover can be realised. Now, as I told you simulated binary crossover basically follow a probability distribution function and so here actually the probability distribution function you can choose any probability distribution function, but it is recommended to follow the two specific probability distribution functions they are basically called $C(\alpha) \wedge E(\alpha)$.

$C(\alpha)$ the probability distribution function is usually followed for contracting crossover whereas the $E(\alpha)$, the another probability distribution used for the expanding crossover. Now, so the probability distribution function that it is followed in case of contacting crossover is shown here. So, this is a function description for a given values of α ; α is already known to us and they it consider q is a constant, the constant can be decided by the user based on the users experience. Similarly, in terms of q if the α is known then this is basically the another probability distribution function this one.

Now, here these are the two recommended probability distribution function, other probability distribution function like Gaussian distribution or some other distribution function also can be followed. Anyway, we will consider these are the two recommend probability distribution function to calculate the children solution following the contracting crossover and then expanding crossover, let us see how these can be done.

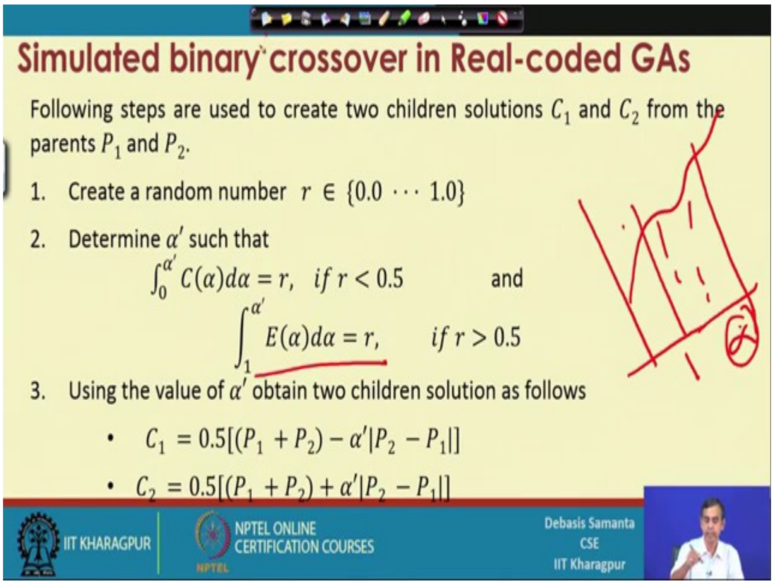
(Refer Slide Time: 19:59)



Now, we will consider the contracting crossover first. So, here basically the three steps are to be followed. So, the first step is that, we have to generate a random number r, that is the random number r. We, this is a first step; generate random number r in between 0 and 1.0, then we have to determine α', how the value of alpha dash can be determined? α' is a value that can be determined using this function.

So, it is basically area, so if this is the probability distribution function and then area covered by these value into 0 and α', so that this area is basically = r. So, if $r<0.5$ and in case of if r, the random number that we have generated here, if it is ¿0.5, then will calculate this one.

(Refer Slide Time: 21:00)

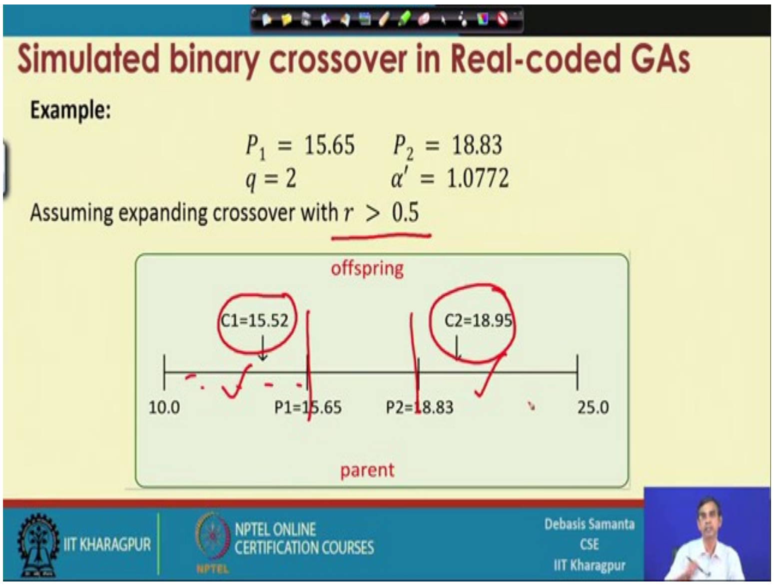


So, it is basically in this case the distribution function it is like this, if this is the distribution function and then it is basically say 1 and then it is basically α', then we will calculate this α' value following this expression, if this one.

Now here, so basically the idea is that the $r<0.5$ the chance that 50% will be belongs to the contracting crossover and then $r>0.5$ that means, another 50% will have the chance to have the expanding; what is called the expanding crossover. So, both way, both expanding and then contracting cross over can be followed to calculate the two values α' according to this distribution function. One α' is known to us, then any two children can be calculated using this formula. So, this is the formula recommended by the developer of simulated binary crossover, this is the formula you have to just follow it.

So, the formula says that is 0.5 and then value of P_1 + P_2 - α' and then it is a what is called the absolute difference between the parent values. Similarly, taking the + sign, the another will be obtained. So, this way the two solution C_1, C_2 can be obtained based on the contracting as well as expanding depending r different values, so this way this they will give. Now, here actually this technique is good, good because we do not have to take any parameters that is required both in linear crossover as well as blend crossover except only the calculation or computation. The computation because this operation or this operation is basically is a computationally expensive operation, but if you able to do it right then all this things are pretty simple and then it is useful and more effective; effective in the sense that it gives better result compare to the linear crossover and then binary crossover technique.

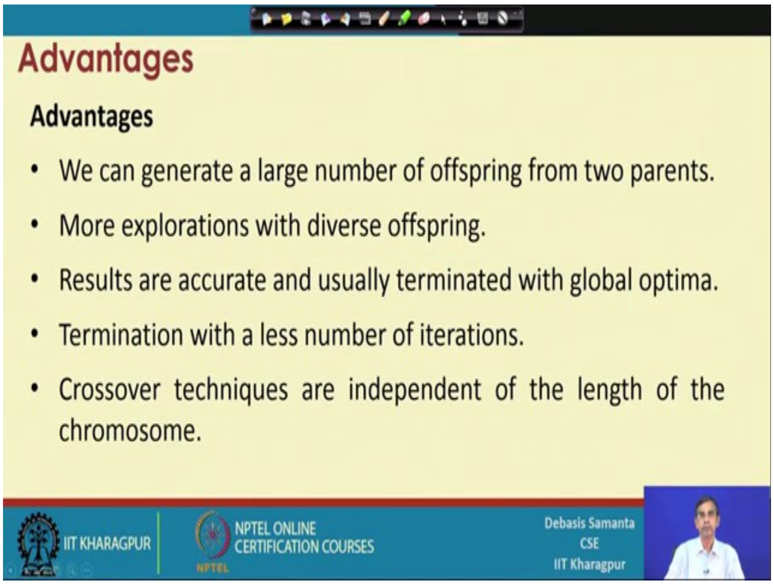
(Refer Slide Time: 22:57)



Now, finally, I would like to give an illustration of the simulated binary cross over technique. So, let us see these are the two parent gene values P_1 and P_2 and we assume $q=2$ is a defined constant that can be varied if you want to have the different results like, it is basically the q value is decided by trial and error, that means, if you take very large values of q whether it is quickly converged or if it is a small value then it can converge with a better solution and all these things. So, here little bit experience of the user is required, usually the user can gather experience by means of trial and error method, that means, they have to run the same program for several cases with different values and for a certain value it will be better result it is taken as that value as the standard value.

Anyway, so suppose $q=2$ known to you and then α' can be calculated based on the random number generation and then the probability distribution function that we have discussed and having these value for example, in this case, so r is 0.5 which gives α' according to the expanding function, then the two values can be obtained, now one is C_1 using the formula that we have already discussed and as I told you $r>0.5$, so it is expanding, that means, this is the P_1 and P_2, it will calculate the chromosome any one region within this region.

(Refer Slide Time: 24:32)



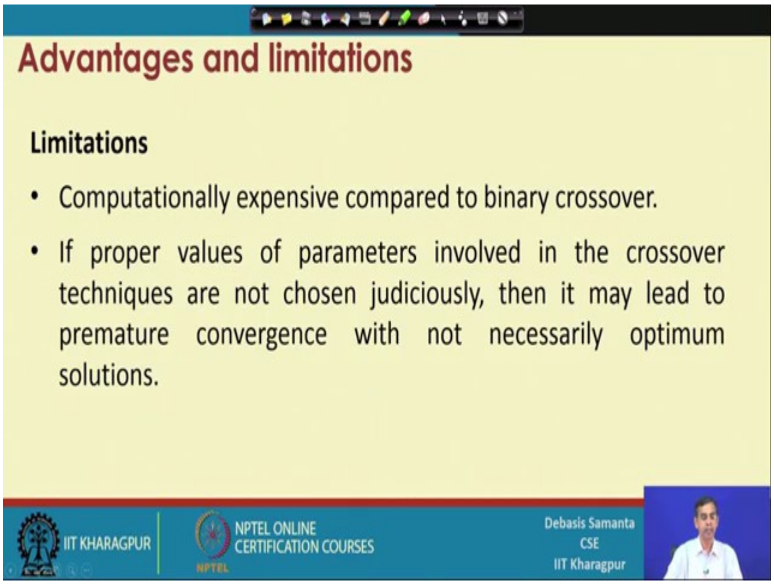
So, this is the idea about the simulated binary crossover techniques that is there and now, simulated binary crossover has a number of advantages compared to the previous two techniques that we have discussed and as in case of linear crossover and blend crossover in this technique also, we will be able to generate large number offspring from 2 parents. So, in that case what we have to do is that we have to generate as many as random number as many we want to have the children's and it in fact allows more exploration with diverse values of offspring, which is comparable to the both linear as well as the binary the blend crossover technique.

Here the results, usually gives more accurate results compare to the linear and blend crossover techniques and usually it gives the global optima, whereas other two techniques usually stuck into the local optima and it basically terminates with a less number of iterations because number of iterations that is required to run the GA is it is in fact, observe that more in case of linear than blend crossover. So, in that sense it is also cost effective, although it is the costly operation in case of crossover, but so far the GA iteration is concerned it require less, so that means, effectively it is a faster GA algorithm than the crossover technique if we follow linear and blend cross over technique.

And here, actually crossover techniques are independent of the length of the chromosome, whatever be the values of the chromosome, that means, the parent values have many number of gene absolutely no problem, we will be able to run effectively using the same techniques.

So, it is fast in that case because the same alpha dash that can be used to calculate the different gene values for the chromosomes, if we take the different this one. So, in the same line we will be able to follow, no need to discuss, no need to consider the different gene values and the different α' and then different random number generation, it is not required. So, in one set the same α' can be used to calculate the different gene values for the offspring for the different parent values or different parent values of the different genes.

(Refer Slide Time: 26:46)



So, this is the advantage of this one. However, it is suffering for another limitation as well as computationally expensive compared to the binary crossover technique. I am now comparing not the linear crossover or blend crossover, but the binary cross over which we followed in case of the binary coded GA. So, if we see, all the binary cross over techniques are very fast and straight forward and pretty simple also, whereas this similar type binary crossover is little bit computationally expensive.

Now, again there is a decision regarding the probability distribution function, if you do not choose the proper probability distribution function or if you do not choose the q values, that is required in case of I mean in probability distribution function decision, then you may lead to an erroneous results and premature convergence. So, user needs to be little bit careful about choosing the right values of probability distribution; right probability functions for contracting as well as expanding function and also the correct values the q that is to be used in the probability distribution function.

So, this is the technique that we have discussed so far the simulated binary crossover is concerned and so we have so far discussed about the two different GA technique, one is the binary coded GA, another is real coded GA and there are several crossover operations. We have learned in binary coded GA and then, just now we have learned about the crossover techniques in the real coded GA. There is another GA encoding scheme which we will discuss, this is the order GA and we will discuss the crossover techniques that is required for the order GA in the next class.

Thank you.

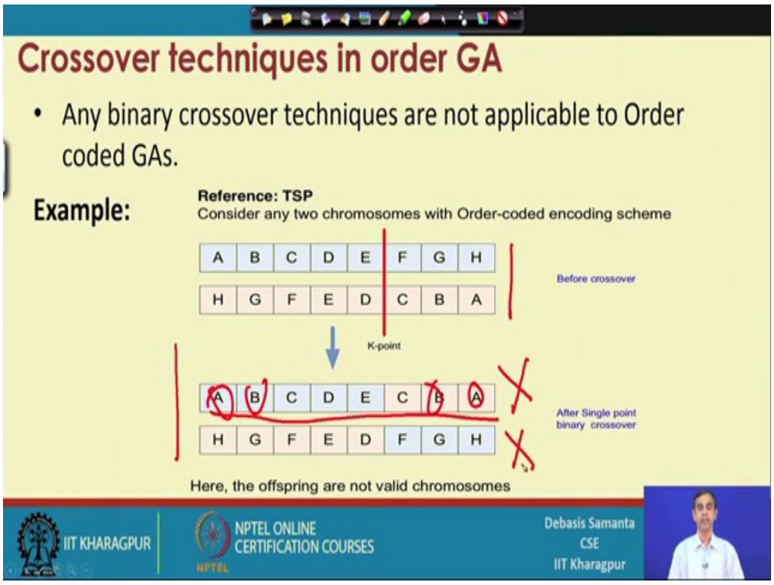
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Science, Kharagpur

Lecture – 22
GA Operator: Crossover (Contd.)

So, different GA encoding scheme follow the different pattern of chromosome. The binary coded GA follow the binary patterns. The real coded GA follow the real values of the gene values and depending on the patterns that it is following binary coded GA. So, the binary cross over techniques were used. We have learned over the binary crossover techniques. Now, real value coded GA, again it is a totally different than the binary crossover techniques because it needs the several totally different what is called the treatment.

Now, we are going to discuss another GA technique. It is called the order GA and the crossover technique that is there in the order coded GA.

(Refer Slide Time: 01:04)



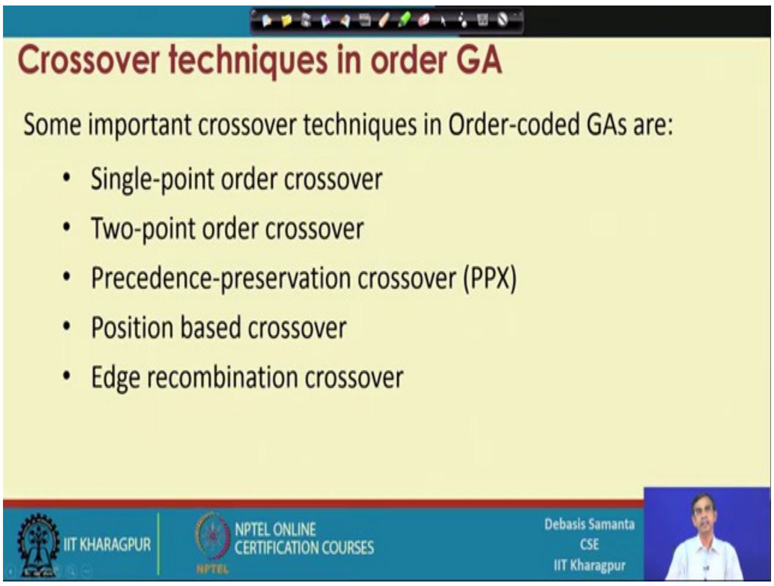
Now, again the order coded GA as we know, it is basically based on the concept of the sequence of the values that is there in the GA. So, the sequence is important. As the sequence is important for example, the travelling salesman problem if we follow that all the values that is there in the chromosome should not be repeated and they should follow certain sequence actually. Now, this means that if we follow the binary crossover techniques; obviously, these are the basically in terms of symbols, so no real values are involved. So, that is why we cannot apply the real coded GA.

However, the binary coded GA cannot, the crossover technique that is used in binary coded GA also cannot be applied here, this an example because how the binary coded the crossover technique that is followed there in binary coded cannot be applied here. Now, if you considered say binary crossover technique namely the single point crossover technique, we can recall that we have to generate a K point there, so if this is the K point, then basically the swapping these two, so it is swapping these two, swapping these two, we will get this one and then swapping this one also will get this one.

So, it is basically from these two parent chromosome using the binary single point crossover we will get it, but you can note that this parent chromosome is not a fusible chromosome or in order to acceptable chromosome because A it is common here, B, B is copied here and all the chromosome that is not all so possibly present here. So, this is not a valid chromosome or this is also similarly not a valid chromosome. That means, the simple single point crossover technique that is used there in binary crossover, it is not applicable to the order GA in fact.

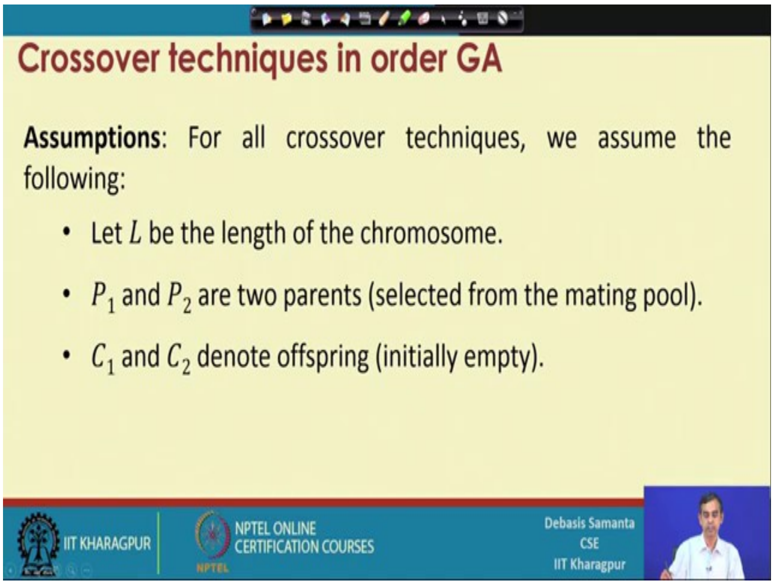
So, this means that order GA needs a different treatment, so far the crossover operation is concerned.

(Refer Slide Time: 03:01)



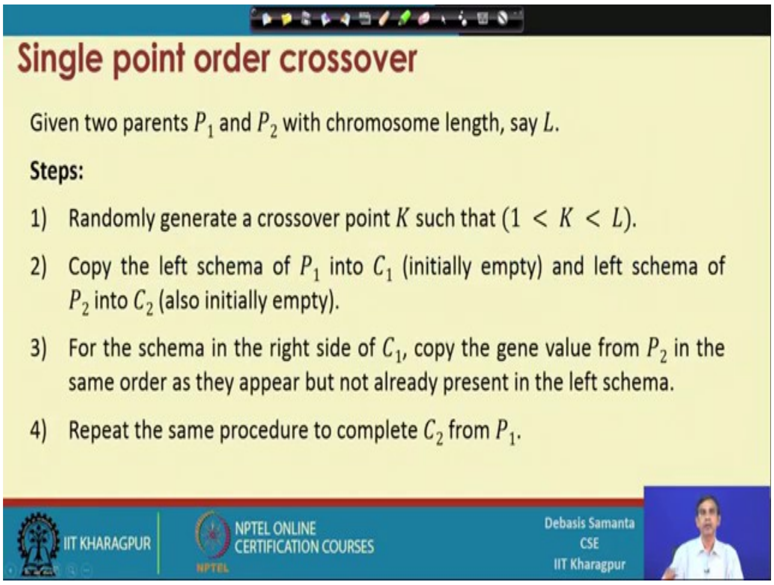
So, we will discuss about the different crossover technique that is followed in case of order coded GA, we have listed few important techniques five important techniques are there, it is one is called the Single-point order crossover, then second is Two-point order crossover, then Precedence-preservation crossover, it is called the PPX, then Position based crossover and then Edge recombination crossover. All these crossover techniques we will discuss one by one in the next subsequent slides.

(Refer Slide Time: 03:29)



Now, in order to discuss the different technique that we have mentioned in the last slides, we will follow certain assumption for all the techniques to discuss. The first is that we will consider that the length of the chromosome be denoted as L . L is an assumption that the length of the chromosome this one. P_1 and P_2 are the two parents which are selected at random from the meeting pool and C_1 and C_2 denotes the two children which we want to derived from the P_1 and P_2 by virtue of the crossover techniques followed there. So, these are the assumptions. Under this assumption, we will be able to discuss each technique one by one. Let us first start with Single point crossover technique in order GA.

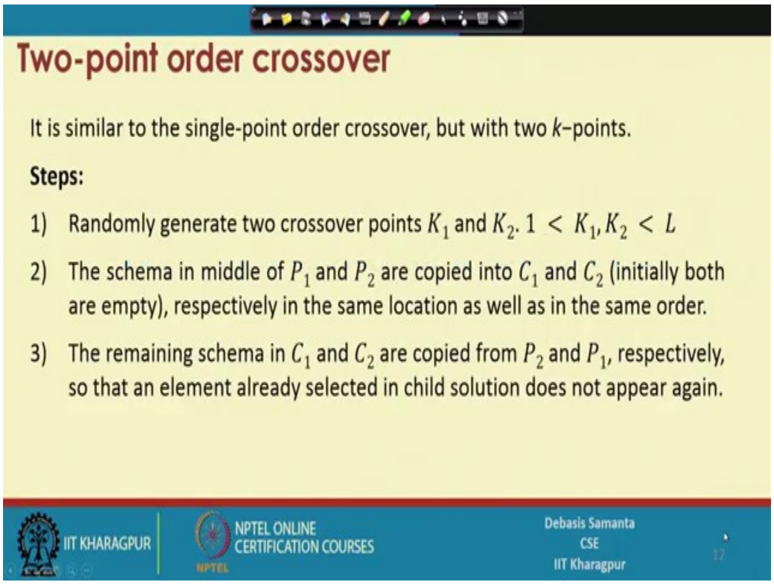
(Refer Slide Time: 04:13)



Now, so if L be the length of the chromosome and P_1 and P_2 are the two chromosomes. Then, in this technique the first task is to generate one number that number should be in between 1 and L and let this number be K. So, it is basically the same as single point crossover is a kinetochore point like. So, K point is to be decided first. Once, the K point is decided this K point is decided then is the next point, next task is we have to copy. So, K point is a kinetochore point that is the point that can define the two parts in both parents, so left part and right part or you can say left schema and then right schema.

Then, the second step, copy the left schema of P_1 into the children C_1. C_1 is initially empty and then left schema of P_2 into C_2 and then for the schema in the right side of C_1 copy the gene value from P_2 in the same order as they appear, but not already present in the left schema. So, if you repeat the same procedure to compute C_2 from P_1, then you will produce the two children solution. So, this is a technique or scheme that is there we have to follow it. Now, let us see how these techniques are there as an illustration, so that you can understand this technique better.

(Refer Slide Time: 05:40)



So, we assume these are the two solution appearance P_1 and P_2 and then a random K point is decided, this is the K point from where the left schema and right schema. So, this party is the left schema and this is the right schema for the parent P_1, similarly this is a left schema for the parent P_2 and the right schema for the parent P_2.

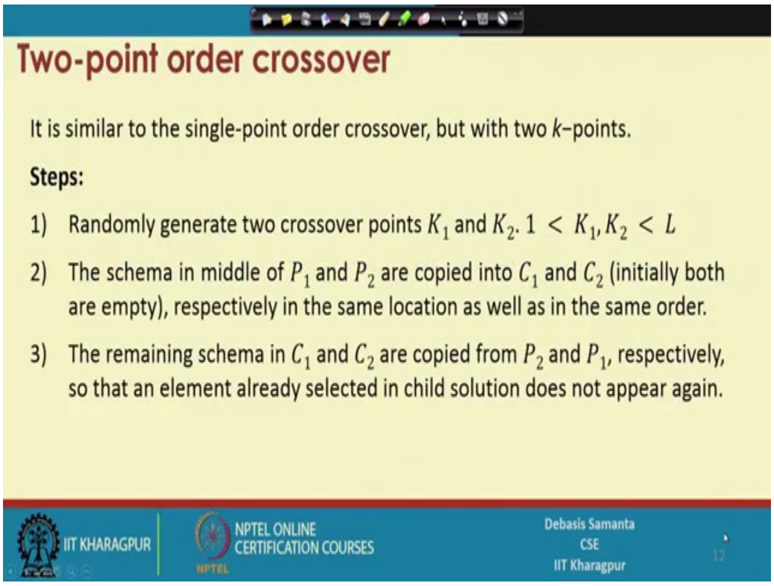
Now, according to this technique, so idea is that, we will copy the left schema from P_1 to C_1. So, it is basically copy, this part is copied to this one and for the rest of the part we will copy from the P_2; from the P_2, but provided that value is already not present in the left part. For example, if we see E, E is already present there. So, E cannot be copied and then D is also present here. D cannot be copied. C is also present here. C cannot be copied. Then, J is not present here. So, J is copied here. Then I, I is not present so far though I is there. The H is there. Now B, B is also not there. So, B is copied here. A is already there. So, A cannot be copied. Then, F is not copied the F is copied and finally G is copied.

So, this way from the parent P_1 and P_2 and based on the kind of a solution C_1 can be obtained. Now, similarly the C_2 can be obtained. In this case, this left schema will be copied to this one and then for the rest of the schema we have to copy it from here right and provided that this is not copied already.

For example, A, A is not in this. So, A will be selected and C is there. C cannot be copied. D is there. So, D cannot be copied. E is here, so E cannot be copied. So, then B, B can be copied here. So, B can be copied here and then F is not copied F is copied there, G is copied there, H is there, not copied. So, I is there and so J is there, J is already there. So, J cannot be copied and I is there this way. So, this way the two chromosomes solution can be obtained and this is the simple technique that is called the single point crossover technique in case of order GA.

Now, we will discuss about another little bit more what is called the diversified technique we can say and it is called the Two-point crossover technique. So, the difference it is basically by it is name, in case of single point we have to consider only one K point, but in case of Two-point crossover we have to consider two points, two K points rather.

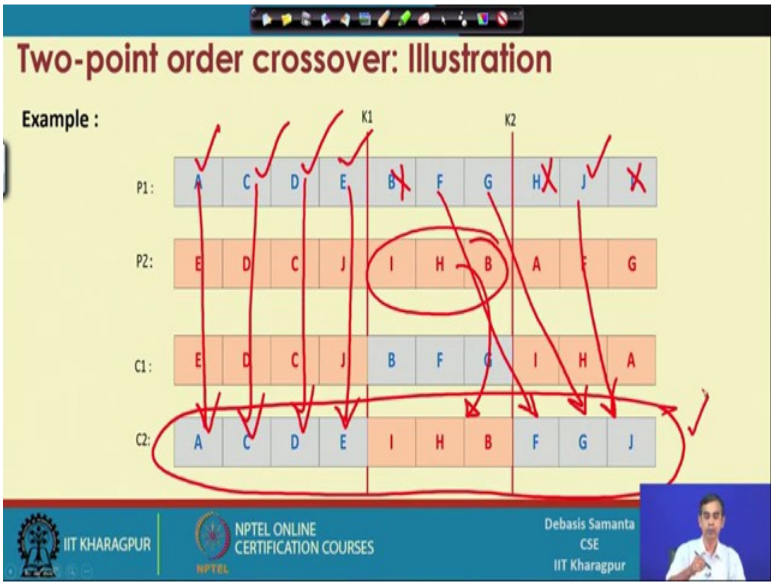
(Refer Slide Time: 08:25)



So, it is just procedure of two, I mean deciding instead of one K values we have to decide two K values and these two K values are denoted K_1 and K_2. The two values are the same as it is in case of the previous one scheme; that means, the values of the K values should be in between 1 and L.

So, once we decide these two K values, then the scheme basically says that the middle of P_1 and P_2 are copied into C_1 and C_2. So, initially C_1 and C_2 are empty, so on the middle part from P_1 is copied into middle part of the C_1 based on these values K_1 and K_2. Similarly, middle part of the P_2 is copied into the C_2 and then once this values is copied, then we have to fill the remaining portions both in the left side as well as right side in both C_1 and C_2. So, it will follow the same procedure as in case of single point order crossover. So, for the remaining point, we have to compare in case of C_1 we compare the gene values from the P_2 and for C_2 we will compare the gene values from P_1 and fill it up.

(Refer Slide Time: 09:31)



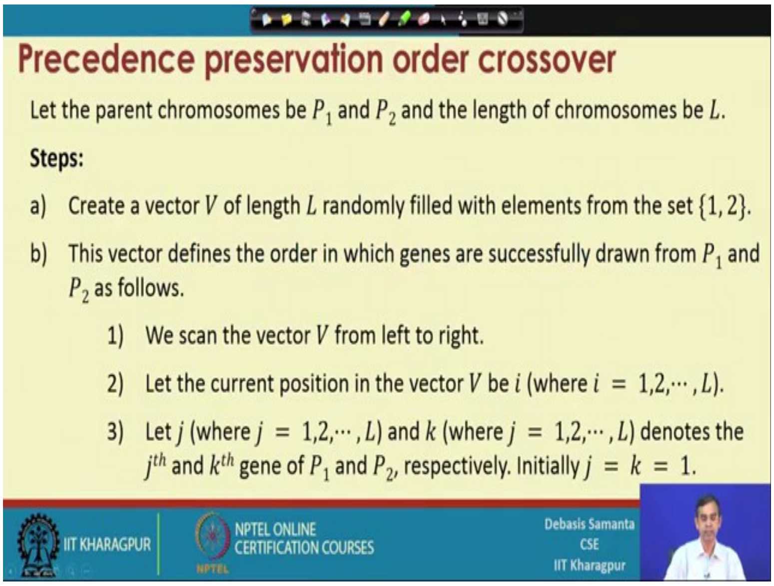
So, let us say one illustration to clear this idea. So, here we will consider two parent chromosome P_1 and P_2 and then two K points K_1 and K_2 are decided at random which is this one. So, in this idea, the idea first is that this is the middle part is copied to the C_1 first and for the rest of the part this one will copy from the parent P_2 provided that it is already not there in the parent; so, in the not there in children C_1 .

For example, so B, F, G already copied, then we will see that E should be selected and E is there. D is also not covered. So, D is selected here. C is selected here. J is also selected here. Then, B, F, G already there. Then, come here I. So, I needs to be selected here because I is not copied. Now H, H is also not covered here. So, H will be here. B cannot be because already B there and then A, A is not copied here. So, A will be there and F and G already there. So it is there. So, in this way, the children chromosome can be obtained.

Now, similarly the C_2 can be obtained. In this case, C_2 will copy. The I, H, B to here and for the rest of the part will copy from here. So, I, H, B is there. So, A should be there. Then, C should be there because C has not been copied so far and D is there. D has been copied and E, E should be copied here. I, H, B it is already there. Then the next part is B. B is not there because B is already here.

So, B is rule out and then if F is copied here because F is not covered and J is copied here because J is not covered and then H, H is already there. So, H is not covered and then J, J is not covered, so J is copied here and then I is there I already, so this A. So, this way the children C_2 can be formed. So, this is the idea about two-point crossover, it is little bit different, then because the single point crossover is pretty simple compare to the two-point crossover, but it keeps the better I am mean diversity in the chromosome solution. So, it is more preferable than the single point crossover. However, this crossover little bit costly operation then the single point crossover. So, next will discuss about the precedence preservative crossover techniques in order GA.

(Refer Slide Time: 11:58)

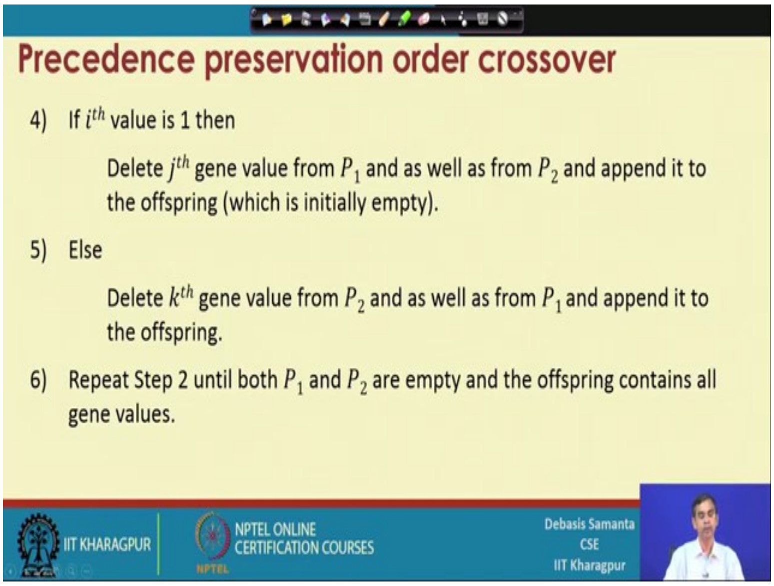


So, we will discuss about the technique here. So, it basically as it is in case of the earlier two crossover techniques in order GA, will follow the two parents P_1 and P_2 and we assume that length of the chromosome be L.

Now, it basically considers one vector, vector with the two different values. Values they are called 1 and 2. So, a vector of same size of the chromosome length L. So, that is why I create a vector B of length L and this is randomly. That means, one vector that can be created with its constituents either 1 or 2 and the length of the vector be L. So, this is basically is called the other pivot one, it is just like a mask in case of binary crossover technique have a half uniform crossover technique that we have discussed in binary coded GA actually, so it is just like a mask like.

Now, then the scheme that is followed in PPX crossover, it has like this. We scan the vector V from left to right. That means, each time we will see whether the current component is 1 or 2. Now, let the current position in the vector V_i that means, we are currently scanning, so it will start from $i=1$ to 1 to the maximum up to L and then j where j is basically 1 to L it is basically a pointer to the first chromosome parent P_1 and k is another pointer to P_1 indicates that at what point of the P_1 and P_2 we are currently traversing.

(Refer Slide Time: 13:40)

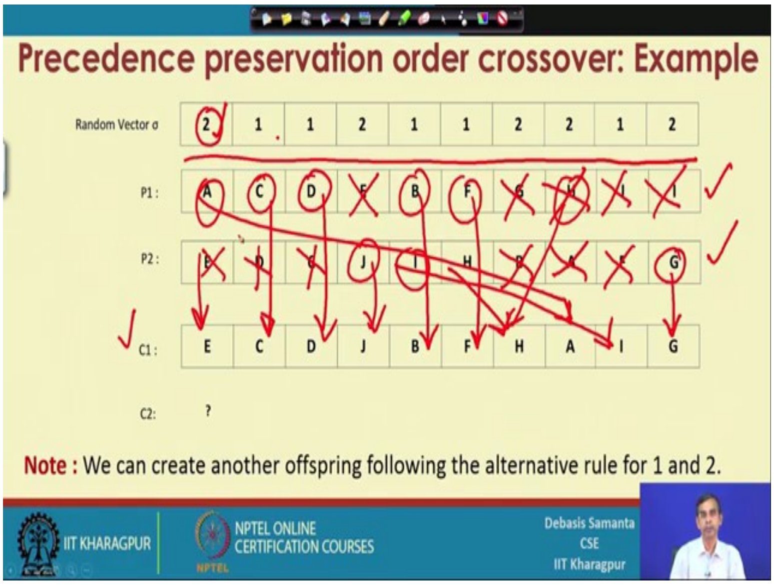


Then, this technique knowing this one, so it basically follows the two method. If i^{th} value is 1; that means, currently the component that is there in vector B is 1, then it basically the idea is that delete the j gene value from P_1 and as well as from P_2. That means, it is select the j gene and then remove this j gene from P_1 and P_2 not to be copied further and then append it to the offspring, which is initially empty. So, it is basically C_1. Suppose, we are considering the creation of children C_1.

Now, if the i^{th} value is not 1; that means, it is 2, then delete k gene; that means, we will just go to the p^{th} chromosome and as well as from P_1 and append it to the offspring.

So, it is basically where they were 1 and 2, it is basically will delete from P_1 and P_2 and then copied into the offspring actually. So, we will repeat the two steps until both P_1 and P_2 are empty and the offspring contents all the gene values. So, better if we can explain the concept of this technique with an example, so here the example.

(Refer Slide Time: 14:51)



So, this is the vector V, with size same as the parent P_1 and parent P_2 and will see how the C_1 can be obtained. So, here the idea is that if P is 2, I mean if the current value is 2 then we will copy from P_2 and if the current value 1 will copy from P_1, so it is like this. If 2 it is there, so this is copied. Here is 1 is there so will copy from this one. Now, when will copy E. So, all E should be deleted both from P_1 and P_2. So, it is been copied there.

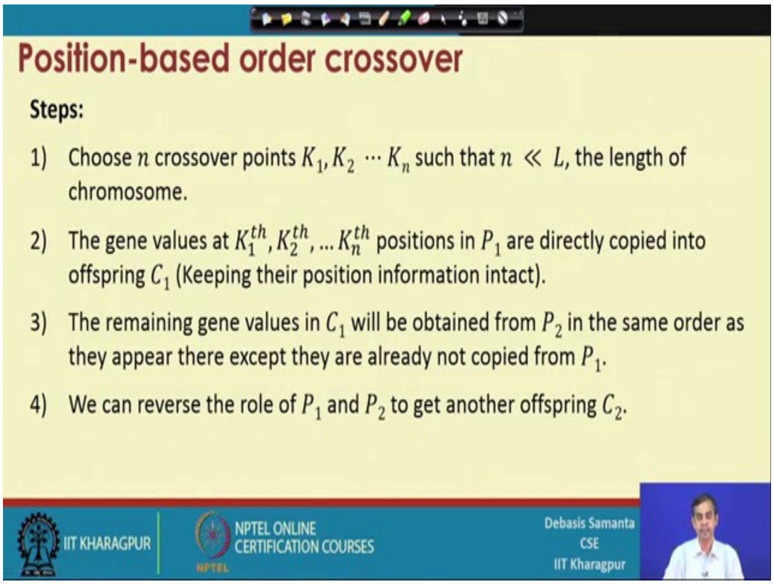
Next, when we wants, so we will copy C and then C as it is already copied, so C will be deleted from the P_2. Now, again 1, so will this one D will be copied and then this D will be removed from there. Then 2, so if it is 2 then will copy this 1 from the parent P_2. So, this is copied here and J, occurrence of J will be deleted from there.

Now 1, so B is copied here and then B is deleted from the parent P_2. 1 so F is copied here and then F is deleted from P_2. 2, 2 G, so 2 means it will be G then, G is basically here. So, G will be copied it from the parent P_2 and then other G occurrence will be deleted from there. So, 2 H, H will be copied from here and then also 2 H H means H is here. So, H will be copied here and then all this H will be deleted from there.

Now, so, then we have to see the 1, 1 means we will copy from P_1, so P_1 will be copied here. So, and then, so this A will be deleted. So, finally, I, so it is 2, 2 means the I is to be copied here and then all other will be removed here. So, this way the entire gene can be copied and then it will produce the offspring. Now, if we reverse the formula policy. That means, if it is 2 then copy from P_2, earlier if it is 2 copy from P_2 then P_1, if it is 1. Now, we revise the policy; that means, if it is 1 then copy from P_2 and if it is 2 then copy from P_1 and then we will follow the reverse one, so the another chromosome C_2 can be obtained.

So, this is the precedence preservative crossover techniques in case of order GA and it was like this.

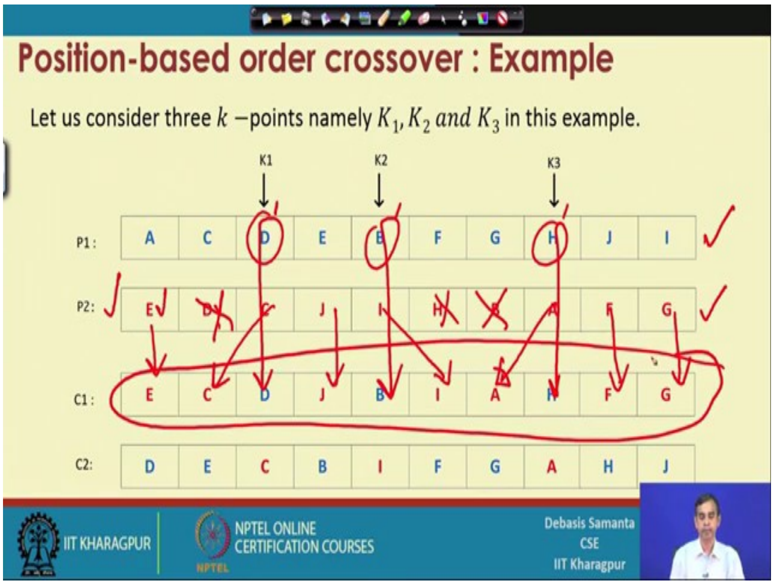
(Refer Slide Time: 17:50)



Now, another technique is called the position based crossover technique, this technique it is more generalized version of the two-point cross over technique in fact. So, here the idea is that choose n crossover points K_1, K_2, … K_n, where n will be sufficiently large than l. So, this is a crossover technique usually followed if the length of the chromosome is too large. So, that we can decide a large number of K points in fact and then, the idea is basically the gene values that K_1, K_2, and then K_n, position in P_1 are directly copied into the offspring C_1 , keeping their position same; that means, K_n value from P_1 is copied to K nth position in C_1 , $K_2's$ values in P_1 is copied to K_2 position in C_1 and so on.

So, this way it will be copied and then so it will get partially filled some chromosome and then for the rest of the chromosome we have to take the confidence of P_2 , we have to take the we have to copy the chromosome values from P_2 provide that they are already not there in C_1 . So, if you follow the reverse; that means the reverse of the previous procedure, then it will give another chromosome C_2 .

(Refer Slide Time: 19:08)

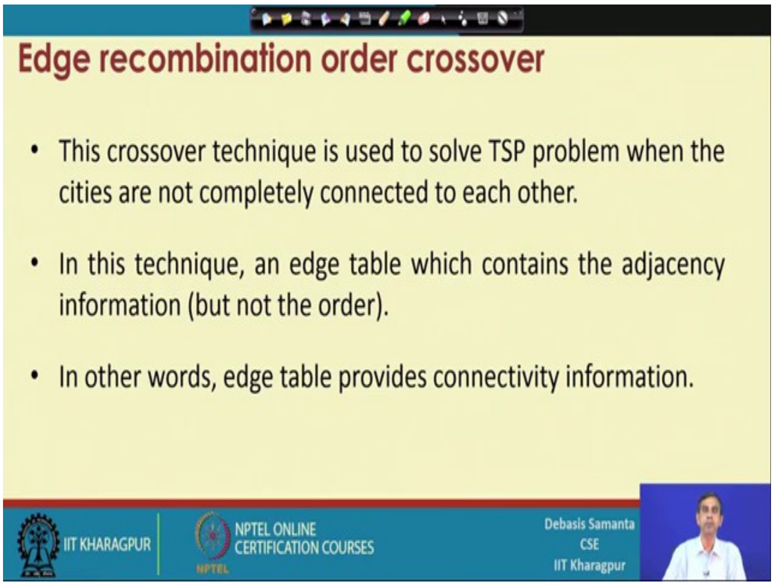


So, let us explain illustrate this technique with an example. We will consider this is the P_1 and another chromosome P_2 and here we consider three points K points, these are called the K_1 , K_2 and K_3 . So, according to the scheme the first scheme will produce a C_1 , so this D one is copied here and then this B one is copied here and then this H one is copied here.

Then, for the rest of the chromosome we have to take it from the P_2 provided already the chromosome which is there should not be into there. Now here, so D, B, A, so H, so will extract the values or copy from the P_2 except D, B, and H which are already there. So, E it is there, so E is coming, D it is there which is not there. So, D is already there, so D should not be here, C it is there because C is not copied there, J it is there because J has not been copied and then I, A it is there, so I, I it is there this I is coming here and then B already there, so B cannot be copied and H cannot be copied because H is already there and A then A can be coming here and F it is not there, so F will be copied and G can be copied. So, this way the children chromosome C_1 can be obtained.

Now, if we follow the reverse procedure in the sense that, if we copy the K_1, K_2 and K_3 points into C_2 form P_2, then will get another offspring for example here, this C is here and this I is here and this A is here. Then for the rest of the part we will copy from there provided that all these things are not there. So, this way you can check it, so these chromosomes can be obtained. So, this is the position based crossover technique there, then the last technique is called edge recombination order crossover technique. Now, edge recombination order crossover is a special case, it is bit computationally expensive, but very famous for the problem like travelling salesman problem.

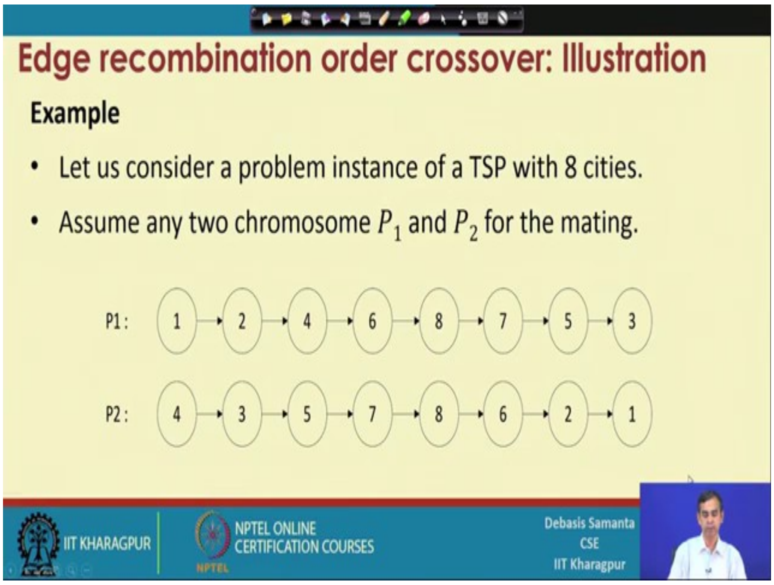
(Refer Slide Time: 21:25)



So, the crossover technique is used to solve the problem like travelling salesman as I told you and so, and also that kind of TSP problem where the cities are not well connected. So, eagerly it works better there and then it is also computationally very fast because the number of chromosomes is basically n factorial and for a large values of n that is really very difficult to find because it is a computationally expensive operation to find the all possible order sequence that is there in possible in the, if the all cities are highly connected.

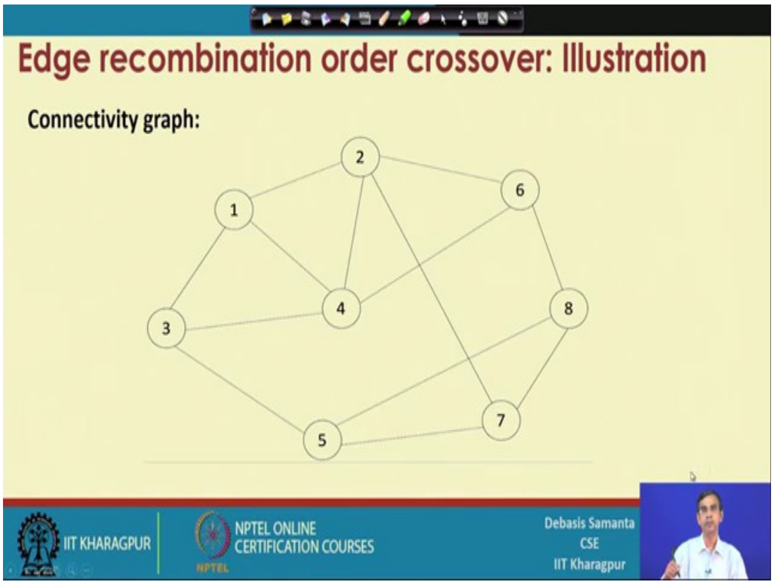
Now, so in this technique basically we will follow on lookup table it is called the edge table, which basically contains the adjacency information and then that atoms is not necessary a particular order in the random order; that means, if a city A, the city A is connected to which of the cities say it is B, C, D, E then they should be present in that edge table.

(Refer Slide Time: 22:35)



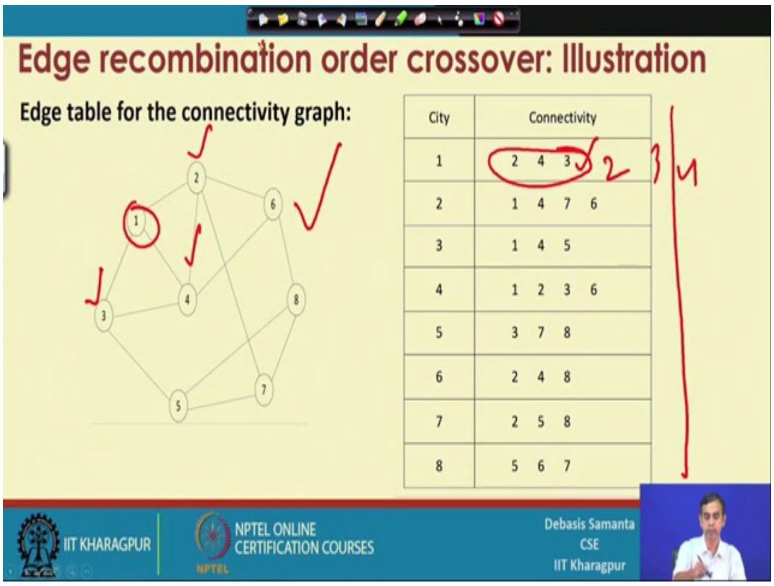
So, then once the, this edge table basically provides the connectivity information in some different form. Now, as an illustration we can consider this problem like. So here, basically the idea is that say suppose these are the two parents P_1 and P_2 for eight cities problem and say you can say that these are the order sequence that is there and this are another order sequence. Now, so we want to find one another children C_1 from these two P_1 and P_2 like.

(Refer Slide Time: 22:59)



So, idea it is basically first we have to create the edge stable and edge stable for a given instance, so this is the problem instance, this basically the showing the connectivity of eight different cities there and as we see that all cities are not connected to all other cities in fact, so there are connectivity's like this. Now, will see for this city map how we can produce the edge table?

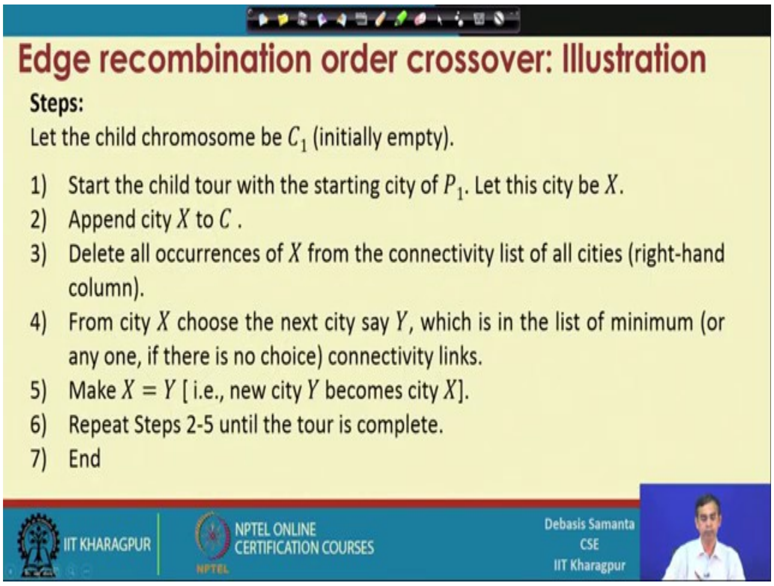
(Refer Slide Time: 23:26)



So, we will see that edge table, so these basically shows the edge table for this city map and as we see for the city 1 we have the three connections 2, 3 and 4. So, we have written 2, 4, 3 and this order is not important, if you say 2, 3, 4 that is also valid. So, the order is not important.

Now, likewise for the city 2 as we see the connectivity as 1, 4, 6 and 7, so it is like this. So, all the connectivities are put it there in the edge table. So, basically idea is that once the city map is known to you. So, city map to know to you, then we will easily obtain this edge table and then this edge table is used for the generation of chromosome for the children.

(Refer Slide Time: 24:21)



Now, let see what is the procedure followed. Here the idea is that, so initially the children chromosome we denote it as a C_1 and initially we assume that it is empty. That means, is blank nothing is there. Then start the children, start the tour with the for the children tour with the starting city of P_1. That means, it is same as P_1 and if we take the starting city of P_2 then another chromosome will be obtained.

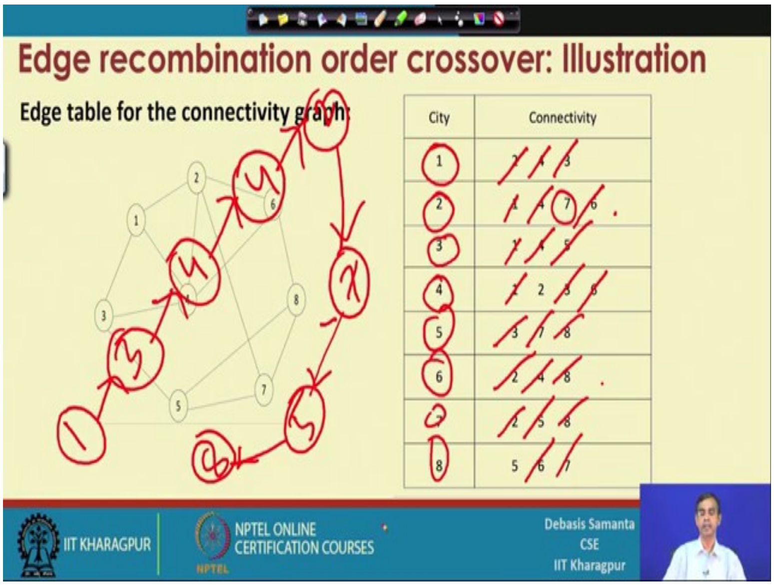
So, let us start with the P_1 first as a parent. So, we will start the; that means, both P_1 and C_1 have the same starting city. It is called the starting city has same as the P1. So, let us denote this city be X. Then we will add this city X to C. That means, this is the first city for the children solution.

Then, once the city X is selected delete all occurrences of X from the connectivity list of all cities that mean as C if the X is selected. So, it should be removed to not to be considered for the others, for the next time. So that X should be deleted from all the connectivity information there.

From city X, then for the city X choose the next city say Y. So, from city X we can and travels into some other city which has the connectivity Y and which is in the, so this the one condition that city X to city that connectivity Y and that also will select that Y because many cities are possible, we will select that Y which in the least of minimum connectivities are there. So, it is like this and then will copy this X to Y and then we make X Y and then repeat the same procedure till we will compete the entire tour for the city; for the solution chromosome C_1.

Now, here is an example that I can tell it. So, suppose the starting city of P_1 is 1. So, we will start from 1 and then from 1 we see that, so 1 it is selected. So, this 1 will be removed, this 1 will be removed, so this is removed because city 1 is selected. So, this is the initially city 1.

(Refer Slide Time: 26:32)



Then, the next city we have to select from city 1 we can go 2, 4, 3. Now, in case of 2 the connectivity is 3, in case of 4 it is again three connectivity that there in case of 3, it is four, so we should select the minimum connectivity that mean 3 in this case, so we will select the next city as 3 and then as 3 is connected. So, we will remove this 3 from every occurrence in the connectivity matrix that mean 3 has been covered.

Then, so we are in the city 3 and from the city 3 we can go to 4 and then 5. So, city 4 and 5 if we go there 4 has the connectivity 2 and 6, whereas 5 has the connectivity 7 and 8 both are same. So, we can take any orbital anyone. So, let it be 4. So, from 3 to 4 we can go to the city 4 and then 4 is covered, so 4 will be removed, 4 will be removed, 4 will be removed and 4 will be removed, so this way. Now, so 4 is covered then 4 from the city 4 we can go to either 2 or 6.

So, we can go anyone, but the thing is that 2 it is a connectivity 7 and 6 and for 6, 2 and 8. So, we will go anyone. So, from the 2, 7 and 6, so 7 and 6 is 2 and 8 and 7 so we can go to from 4 actually 2 and 6. So, 2 has the connectivity 7, 6 and 6 has the connectivity 2 and 8. So, we can go anyone, let it be moved to 2 right, so 6, 1, so 4 to 6.

So, 6 is connected and then 1, 6 is connected it will remove the 6 from every occurrence it is there, from the 6 we can go move to 2 or 8. So, 2 has the only 7 and then 8 has 2, so we can go to 2. So, we can go to 2 to from 4 to 6 and then 2 will be removed from here and there. So, from 2 we can go to the 7 finally, so 7 is there, 7 has this 5 and 8.

So, it is 7, 7 and 5 and 8 out of this 5 and 8, 7 is deleted. So, 5 and 8 it is there we can go anyone, so it is anyone maybe 5, say 5 and finally, so 5 is deleted and finally 5 to 8, so the 8 into there. So, this basically gives the children chromosome according to the edge combination technique. So, this way you can follow it. Now, as we say, that so total tour is completed and covering all cities there. So, this is basically the idea.

Now, here we have started with the starting point of city P_1. Now, if we follow the starting point of city which is P_2, let it be say 4 then definitely it will produce the different one chromosome, so that will be considered as C_2. So, this way so P_1 and P_2 chromosome influence to obtain the two children solution C_1 and C_2 according to the edge recombination technique.

So, we have learned the different crossover techniques related to the different kind of GA encoding scheme, binary coded GA, then real coded GA and order GA mainly. These are the three different GA techniques are very popular. So, we have learned all the crossover techniques and what is want to say is that crossover techniques are the most important and the significant operations out of all GA operations are there, like in encoding and selection. This is because the crossover we have to follow from the np number of mating pools to create so many chromosomes. That means, it is to be computed maximum.

Therefore, while we are choosing the crossover technique, we have to choose that which takes the minimum time to compute because the overall efficiency of the geo-technics depends on how fast we can accomplish the crossover operations. So, these way crossover operations are very vital, one operations in case of GA algorithm and we have discussed the different operation techniques so far. Our next portion is basically the mutation will discuss in the next class.

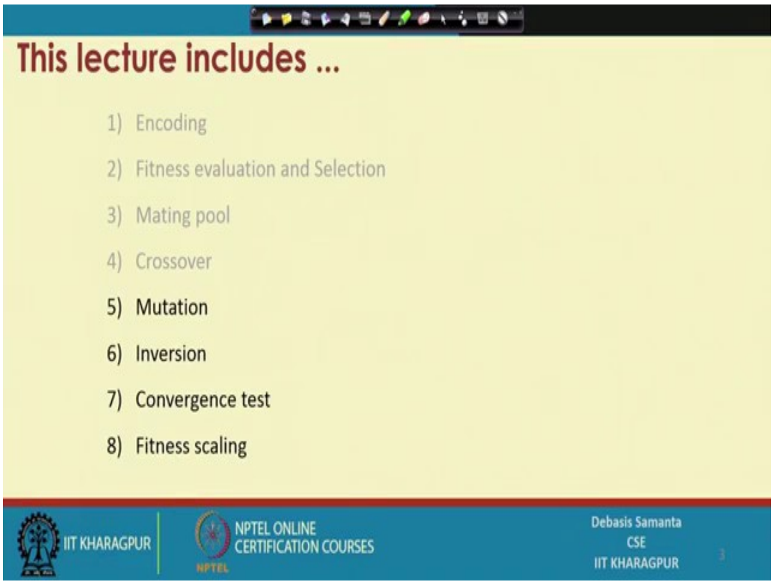
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Science, Kharagpur

Lecture – 23
GA Operator: Mutation and others

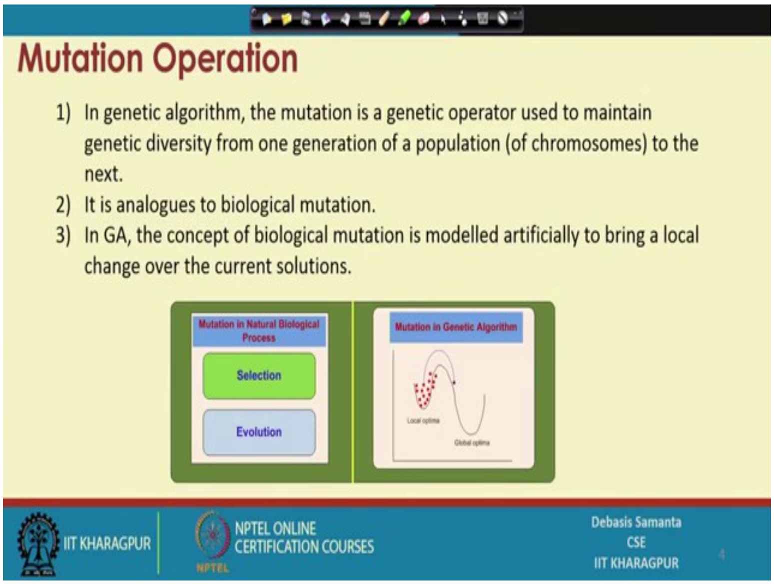
We are discussing operations related to genetic algorithm. Today we will discuss few more operations. So, this is related to reproduction task. The operations are mutation, inversion.

(Refer Slide Time: 00:36)



So, like the crossover operations that we have discussed and it depends on the different type of encoding scheme. That means, different crossover techniques are to be followed to different type of GA. So, here also the mutation operations are dependent on the type of encoding scheme that we are following.

(Refer Slide Time: 00:56)



So, first let us discuss about why the mutation operation is required? Or in what context the mutation operation in genetic algorithm is significant?

So, basically sometimes whenever the GA is in certain intermediate stage of iteration, it appears that there is not so much desirable level of population diversity. As a result, whatever the solutions are obtained, they lead to a solution final solution because there will be no scope of further improvement. Now in order to search bit in a different direction so, what exactly the thing it is required is that, we have to change some chromosomes, some solution in an abrupt manner. So, the way it can be solved, I mean it can be changed and this is called the mutation operation.

Now, the mutation operation is very much similar to the biological mutation, and we know in case of biological mutation. So, there is a all of a sudden changes in the gene for example, in your garden if you plant a rose, tree and then the flower of the rose tree it is say it is red, then one day all of a sudden it is quite possible that one flower which appears is basically white colour or there is a tree out of so many red rose tree, there is a one tree which produce say the white flower.

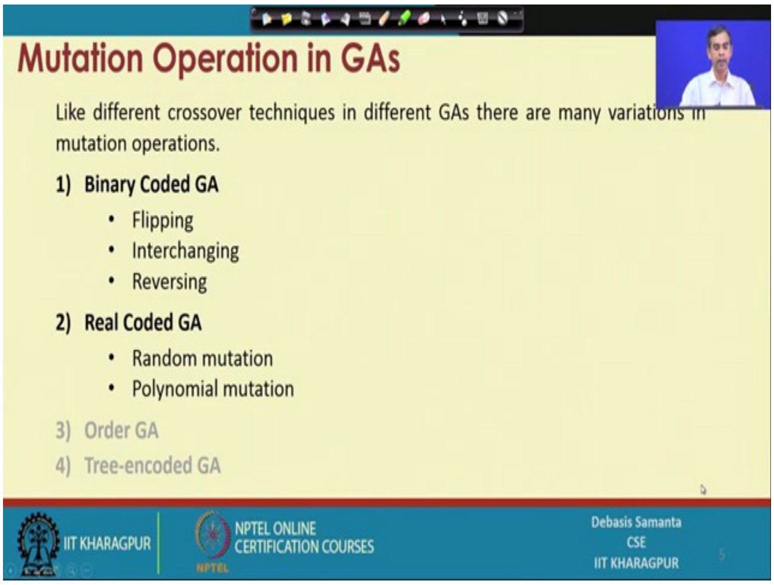
So, this is an example of mutation in our nature. So, similar mutation is followed in case of genetic algorithm also. So, basically it is the idea about that how forcefully we can modify a chromosome. Now why this mutation operation is required? We can understand these things if we look at the figure. So, this figure is basically planned for this purpose.

Now, we are searching for a solution and suppose at any instant the solution that we are here. So, if we do some what is called the reproduction by means of cross over and everything. So, from this two solution; it produces another solution here, so all the solutions are confined into this one, then after some iterations if we do not find any improvement, then we can accept the solution as the optimum solution. In fact, this is the solution that we obtain; it is called the local optimum. This is because we can diversify the such phase from this region to some other region here or some other region here so, that we can find, if some other what is called the optimized possible.

So, if we can change some chromosomes from this region to this region, then we can come to this another optimum maybe the better optimum or the global optimum. So, how these sudden changes in chromosome can be takes place? For example, if we consider this is the one chromosome and if we mutate it, then the mutate chromosome this one, it can lead to I mean to your search to some other optimum value.

So, this way the mutation operation is very much effective, and it is usually applicable when before taking a final decision that weather we will terminate the de iteration or we have to do something. So, in that case we can forcefully call the mutation operations on some chromosomes, and see whether the mutated chromosomes when solution is mutated chromosome, it can give better solution or not. So, this is the one way that the population diversity can be achieved and the mutation operation is mean for this purpose.

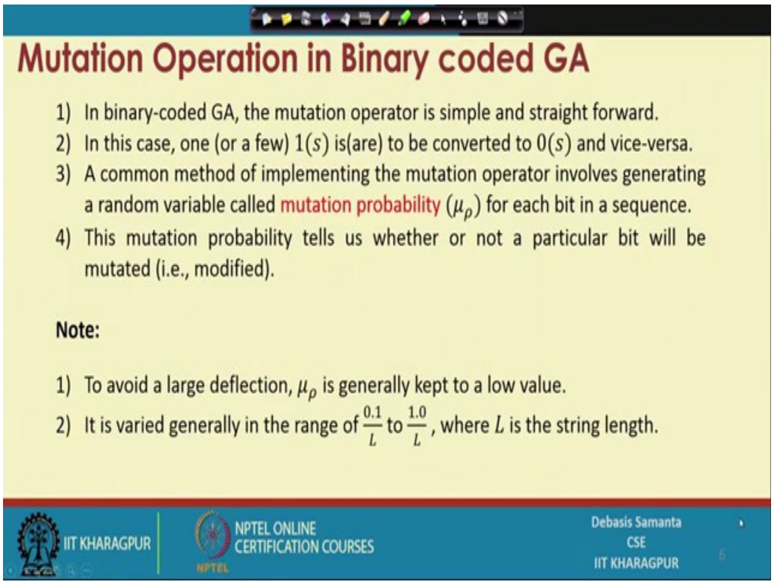
(Refer Slide Time: 04:56)



So, this is the a rational behind mutation operation, and there are different operations related to the difference GAs and in this lecture we will learn two different GA type mainly the binary coded and real coded GA. The other two mutation operation in other two GAs like order GA and Tree-encoded GA is left as a self-study. So, it will not be possible to cover because of the timing constant.

So, we will discuss about two GA those are basically more important people usually follow binary coded GA or real coded GA. So, we will discuss our idea about these two. So, in case of binary coded GA. So for the mutation operations are concerned, there are three different versions one is called flipping, interchanging and reversing. Similarly, for the real coded GA the mutation operation two different techniques are followed one is random mutation and then polynomial mutation.

(Refer Slide Time: 05:52)



So, first let us discuss about the binary coded GA mutation, and we will discuss the flipping operation first. Now before going to take discussion about the flipping operation or other GA operations, mutation operation in binary coded GA; we will tell exactly what is the basic concept that is followed; so far the mutation is concerned.

Now, the mutation operator basically we know in case of binary coded GA, chromosome is represented in terms of 0's and 1's. So, if we change some 1's into 0 and vice versa; then it is called the mutation. So, basically changing some 1's or 0's to 0's or 1's and it is a mutation.

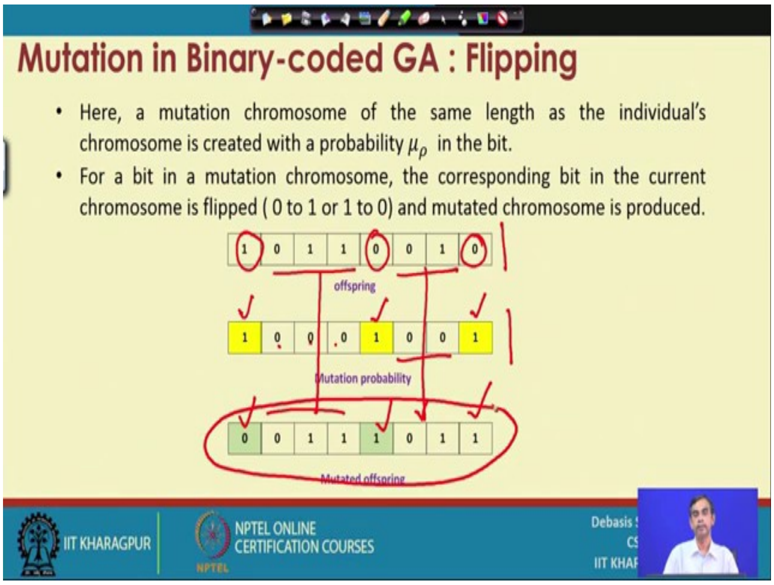
Now the question is that which bits in the binary chromosome should be changes? So, to decide it, we have to follow the mutation probability. That means, for each bit position; we have to decide the mutation probability, and then based on this mutation probability, we basically decide that how many bits are to be fit, or how many bits are to be reversed or interchanged?

So, this is the one parameter that user has to decide the mutation probability. So, we denote this mutation probability by the symbol μ_p, and there is a heuristic; heuristic is that this μ_p should be a very low value. If we take the high value then your search will be in a random direction, that is some time not desirable and then it will take longer time to terminate the searching process and you may not get the optimal solution at all.

So, there is a good heuristic that is usually followed in genetic algorithm is that, if L is the length of the chromosome then this mu rho should be within the range $\frac{0.1}{L}$ to $\frac{1.0}{L}$. So, if the value of L is large as we see the, this probability will also reduce into a smaller value like.

Anyway so, this is the heuristic that is followed, and this is the heuristic that is followed to decide the μ_ρ it is called the mutation probability. So, like different parameters mutation probability is also another parameter it is called the GA parameter to be decided by the user during the execution of the genetic algorithm.

(Refer Slide Time: 08:22)



Now, having this is the concept mutation probability, we will discuss about the first operations related to the mutation in case of binary coded GA. The operation is called the flipping and as the name implies flipping means the 1 will be flipped to 0 or 0 will be flipped to 1 it is kind this. Now first I told you the mu rho needs to be decided so in fact, we will toss a coin and we will decide the coin in such a way that, mu rho number of tosses will be 1 and other will be 0.

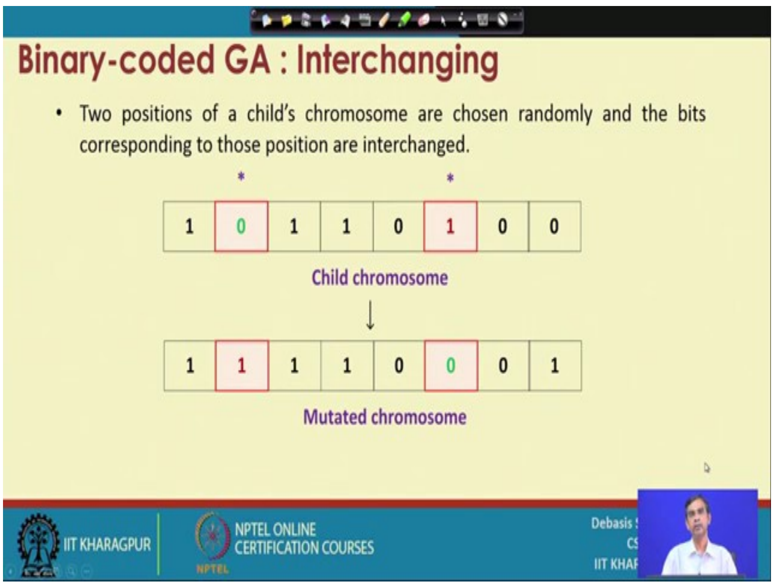
So, somehow this toss can be planned or some program can be written. So, for example, this is the toss. So, this toss is decided in such a way that only few of the bits to be flipped. The bits which are to be fit it is marked as a yellow colour. So, this bits, these bits and these bits.

So, if it is like this then suppose this is the one chromosome that is given to you we can say the child chromosome are offspring chromosome.

And so, based on this mutation probability we can flip for example, it is a 1. So, this flip will be flopped. So, it is 0. Now all the 0 0 and 0; so, this will remain unchanged. Now again 1, so this 0 will be flipped to 1 and there is a 0. So, this means they will not be changed and here this is 1. So, it is 0; so, this will be 1. So, this way this is the mutated chromosome can be obtained after the operation of flipping. So, idea is pretty straightforward and simple; in fact, and it is also not so time consuming operations.

And generally the child chromosome we have selected at random; that means, which chromosome needs to be mutated. So, you can take the child chromosome at random. And not all the child chromosome to be mutated, there are again few I mean child chromosome to be mutated. So, that is also very lesser number of things to be mutated, and it depends on. If you want to have a very high diversity, then you can go for a large number of child chromosomes and for a smaller diversity it is less number of child chromosome can be considered for the mutation.

(Refer Slide Time: 10:40)

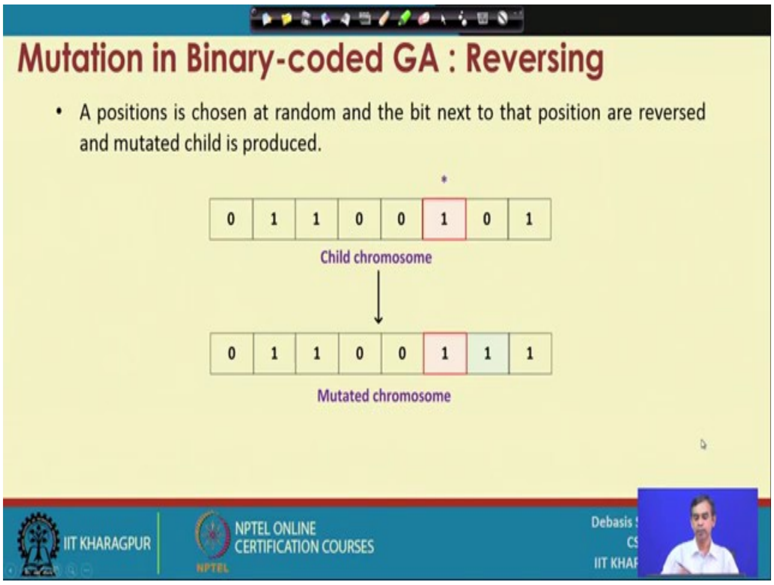


So, this is the flipping operations, and I will discuss about the next operations mutation operations it is called the interchanging. So, in this case again two-bit position are to be selected at random. So, randomly we select two-bit position for example, this is the child chromosome and then these are the two bits are selected at random in between the inter chromosomes, there and then interchanging means this bit 0, if it is 0; it will be interchanged to 1 and if it is 1; this will be 0. It is just like a flipping of course, but in case of flipping operation, there are certain tossing required.

But here we do not have to do any tossing only the thing is that we have to select two-bit position random. Sometimes instead of two bit positions we can take 3 bit positions or more number of positions, and then accordingly all those bit positions can be mutated.

So, it is almost similar as the flipping operation also, but let it be in a different way. So, user can try with first two bits to be mutated then three bits, and then less number of bits whatever it is there and this one. So, it is more controllable than the previous one in fact.

(Refer Slide Time: 11:56)

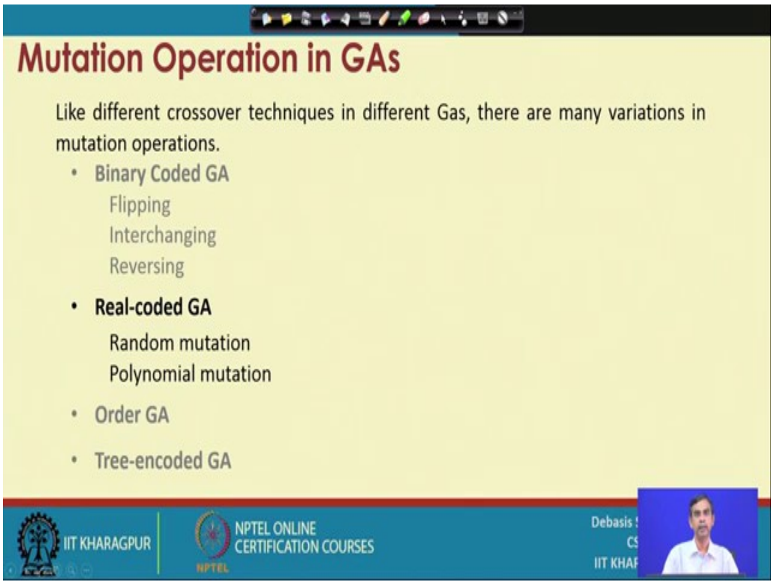


So, this is an interchanging and the next operation is called the reversing. So, idea it is basically the idea it is that, here in the previous interchanging operations what we have to do is we have to select k number of this position, but here you do not have to select k number of bit positions, this is basically the mutation operation is required whenever you need a very slight modification in the chromosomes.

So, it is basically the idea it is that you have to first generate a random number or we can say the decided a bit position at random. For example, this is the child chromosome, and we decide one-bit position at random this one. Then what is the procedure here is that either the previous bits or next bit whatever it is there. So, we can fit it. So, the previous bit for example, the next bit suppose if you consider the next bit 0, 1 up to the selected bit will flip it. So, if we flip it then it will give this one 1. So, this is a mechanism that is followed here in case of reversing, and this is again most simple method compared to the previous two methods that we have discussed.

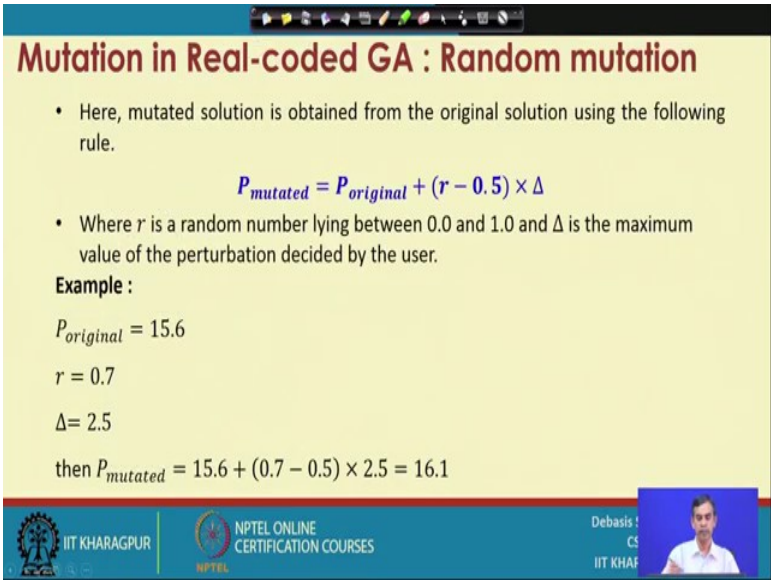
So, these are the three methods that is followed in case of binary coded GA and for the mutation operation.

(Refer Slide Time: 13:10)



Now, we will discuss about the real coded GA, as I told you there are two techniques the random mutation and polynomial mutation. Now let us discuss the two techniques one by one. So, first we will discuss about random mutation.

(Refer Slide Time: 13:25)



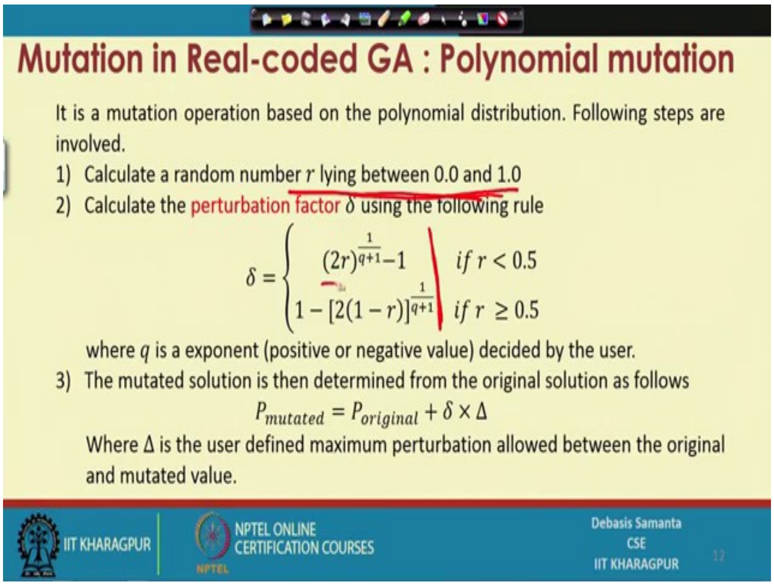
So, here mutation solution is obtained by means of some what is called the formula, this formula is basically given empirically, the formula takes the form which is shown here. So, this is the formula. So, it basically the parent values of the original it is basically child chromosome, and then it computes some calculation like this here the r is basically the random number random number in the range between 0 0 and 1.0, and Δ is also another constant it is decided by the user. So, this constant it is basically called the perturbation factor.

So, if we compute this equation, then it will produce a mutated chromosome that is denoted as $P_{mutated}$. So, in this operation the ask that is required you have to generate a random number first, and then Δ is already predefined constant decided by the user and use this calculation this calculation is also simple calculation only in terms of some division, addition and multiplication not so, costly calculation here and we will be able to obtain the another value from a given value.

So, idea it is like this let us illustrate the concept of this random mutation with an example, say suppose this is the parent value I mean chromosome value of a child chromosome and this is the random number, which is generated at that instant and this is the fit factor that is decided for this process. Then the mutated chromosome can be represented which can obtain this value, if you follow the expression which is already stated there. So, this way from one value that is for the chromosome value belongs to a particular child will be mutated to another value. So, there is a slight change basically we have now; obviously, much how much deflection, how much diversity you want that depends on these factor. So, we can control these values, and then accordingly some values can be obtained which is higher than the value that is required. So, these are the basically perturbation factor, if we control this value then the different chromosome mutated chromosome can be obtained.

So, this is the idea about that random mutation.

(Refer Slide Time: 15:52)



Now, we will discuss about a little bit computationally expensive, but gives better result usually it is called the polynomial mutation.

Now, in this mutation like in case of random mutation we have to follow a random number. So, this is a random number r in the same range as in between 0.0 to 1.0 and then we have to calculate another factor here, it is called the perturbation factor Δ and. In fact, in the previous method it is user's responsibility to decide the Δ .

But in this case it is an idea it is given that the delta can be calculated more statistically or more probabilistically then, and this calculation is based on some statistical function distribution function which is there. So, one distribution function is follows there, if the random number $r<0.5$ and another function that is followed here if the number less than the, this one. So, these are the two expression given by the developer or the designer itself and we can consider this as empirical formula and then.

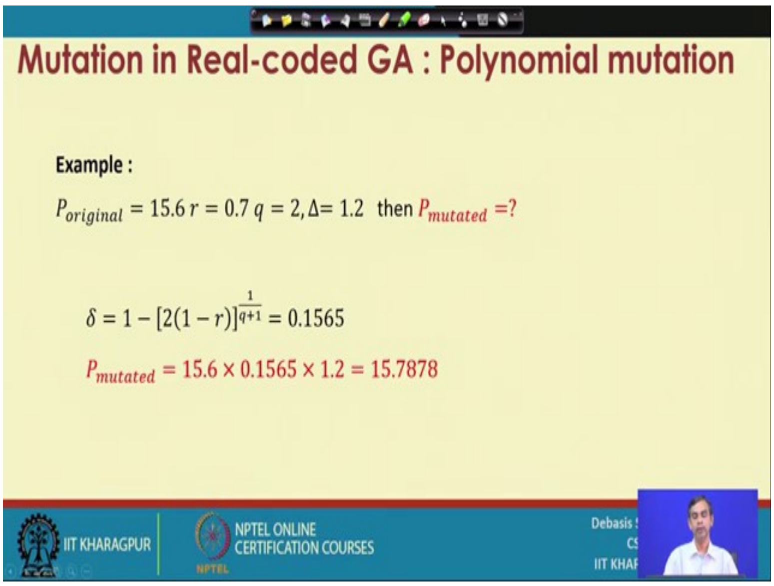
So, following this formula and based on the values of r we will be able to Δ. So, here we can see in the previous case Δ is fixed for any operations, but here the Δ is not fixed rather Δ is decided by the r value always. So, here because r is there and r is there and accordingly this one. So, Δ is basically dependent on r. So, delta is not truly a fix for all mutation operations for any other chromosome. So, it is basically varies from one operation to another operation as r varies.

And here in this operation another constant to be decided by the user q, like Δ that is there we have to also decide one constant and that constant can be based on the designer experience or users experience.

So, once you know the value of q, and then r can be decided a random and accordingly Δ can be computed, and then we will be able to use this formula here. So, the mutated chromosome that can be obtained using this formula. So, $P_{original}$ and Δ and this is again another perturbation factor, that we have followed there in case of random there also you have to concerned it or sometimes only Δ something into this one some other in terms of values also can be constant. So, anyway. So, if we fix we consider a fixed deviation that is allowed. So, this is the delta and then based on these things the mutated chromosome can be obtained.

So, this is the idea about the polynomial mutation in case of GA real coded G A.

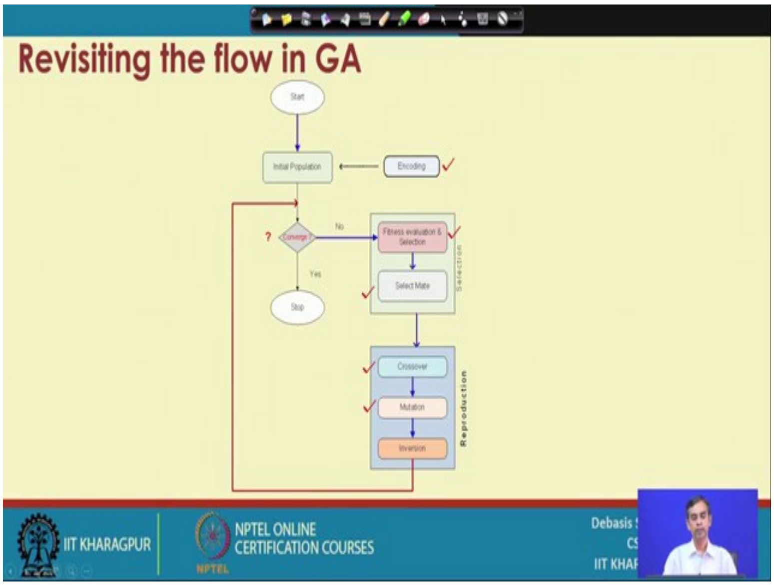
(Refer Slide Time: 18:34)



Here is an example in this example we consider the child chromosome the value is 15.6, r is decided a random 0.7, q is the 2 standard constant data is another constant remain throughout this one. And then now we have to calculate $P_{mutated}$. First we have to calculate δ in this case r is 0.7. So, the second formula needs to be followed and this formula gives the value of δ is this one and once the δ value is known we will be able to calculate using the same formula and then this one. So, if this is the child chromosome then the mutated chromosome is like this one. So, this way the mutated chromosome value can be obtained.

So, these are the two I mean techniques that is there for the real coded GA.

(Refer Slide Time: 19:24)



Now, we have discussed many operations and the GA regarding the GA cycles. We have discussed about the how to create the population by means of the encoding scheme, and then we have also discuss about how to evaluate the fitness of each solution and then the selection can be carried out, and then we have to create a meeting pool. So, this completes the selection operation.

And then comes to the reproduction operation. So, for the reproduction operation the crossover and mutation that you have been discussed in details. Now there is another operation, it is basically called the inversion. It is part of the mutation operation. It is part of the reproduction task.

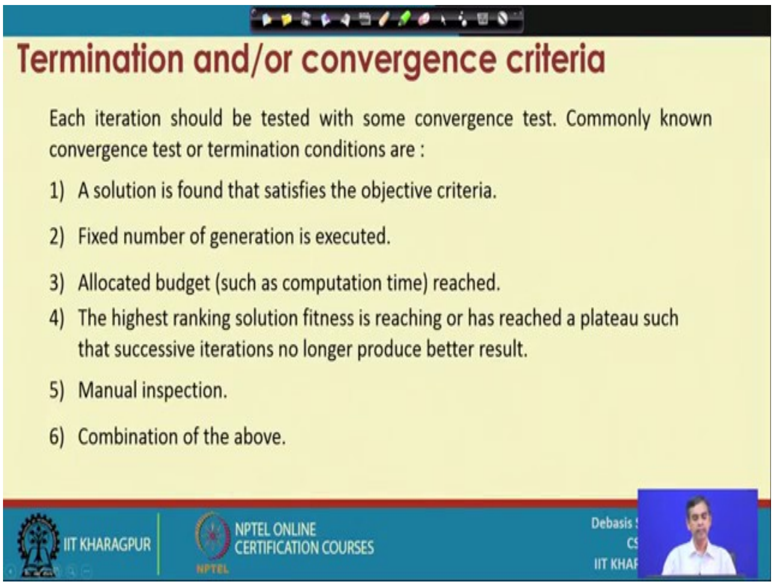
So, in case of inversion operation is a very drastic one operation usually occurs very little time in the entire GA cycles maybe out of the 100 cycles we have to follow 1 or 2, and that is to not to all chromosome for some chromosomes. So, the inversion operation is basically select some chromosomes in the current population at random. Say out of thousand we can select may be say 20. So, 0.2% like this one or this one, then out of this a selected chromosome in the current population we have to follow inversion before going to either crossover or mutation or cycle basically.

So, inversion operation basically it will change in case of binary coded, it will change basically all 0s to all 1s and all 1s to all 0s. So, this way a drastic change can take place on the other hand in case of real coded GA if the value is very low then we can change this to a very high value. So, low to high value or high to low value is the inversion operation that is in case of real coded GA.

And so, these are the operation that is there. We have discussed all these operations. Now, here we have to discuss about the convergence. That means, how to terminate this cycle or how long we have to continue this searching for the optimum solution?

So, in the next few slides will discuss the convergence operations.

(Refer Slide Time: 21:40)



Usually the convergence criteria we have to follow. I have listed few important criteria usually the GA programmer follows. So, the first criteria say that the first criteria is basically if we find a solution, which is basically our expected one solution then we can terminate because if we know that this is the expected results to be then we can stop it there and then what are the solution we got it design values design parameters result we can take it as a solution.

So, this is basically whenever you expect the desired result satisfying the objective criteria then we can stop it there. This is a first criteria that we follow, in the second criteria is basically we can define the maximum number of cycles that we should execute. So, is basically how many cycles that needs to be executed if we decide, say maybe it is 50 sometime 100, depending on your I mean computation affordability how much computation time that you can afford we can decide the fix number of cycles that needs to be iterated and based on this things second criteria is followed.

Now, another idea is that the budget allocation. Budget allocation in the sense that I will allow maximum 3 hours, to run one GA algorithm. So, 3 hours in the 3 hours if it is terminating before this thing it is fine if it does not we have to continue the search till the 3 hours is over. So, depending again based on the programmers time available.

So, they can fix the budgets that mean computation time budget or sometimes the memory budget also. So, within this memory we have to solve it, then how many iterations whatever you want you do it. So, it is whatever the budget it is their time budget or memory budget we have to follow it and then so long time budget permits we can cycle the GA operation.

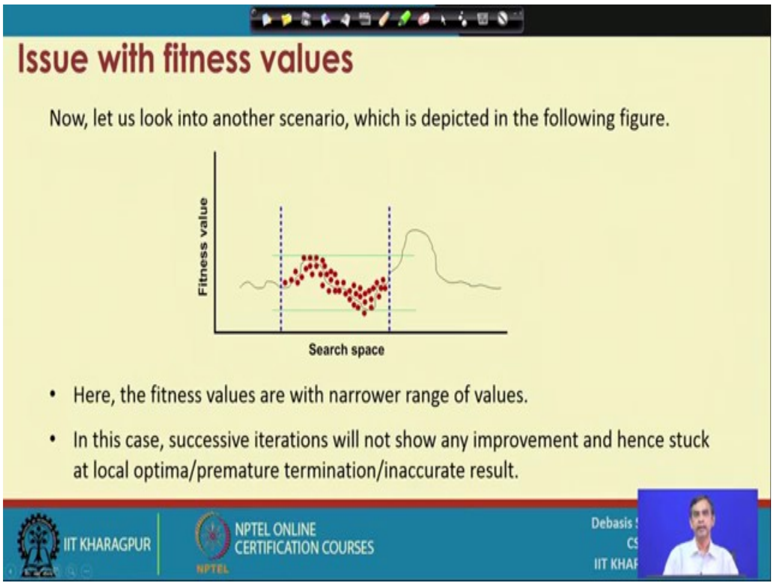
The next is. So, another criteria is that. So, fine sometimes we have to find the best solution, best solution in terms of say ranking of the fitness if we find one highest ranking fitness solution right that has reached to a basically after a successful number of iterations. So, after suppose 10 conjugative iterations, we are getting the highest ranking solution all the time. This means that we have already come to a global optima and then we can stop the search criteria there. So, this is the one criteria and then manual inspection. It is a little bit tedious and very difficult. If it is very small number of solutions are there then we can do it. So, is basically.

So, check the solutions one by one if you plot the solution graphically also sometimes it works, and then we can decide whether we should terminate this one. So, in this case you have to run on cycle check the solution manually and then decide whether you should continue the iteration or stop it. So, this is; obviously, not a desirable operation many programmer do not like it, and another criteria is basically combination of anyone or any two or any combinations out of the five criteria that we have decided. So, this is obviously, at the cost of time because we have to check lot of things in after every iteration, because whether these are the criteria is satisfied or not.

So, this is basically the rule of thumb. So, for the convergence criteria is concerned and usually we follow this kind of method.

Now, so, we have learned about the different operations in particular operations related to reproduction. Now there are few issues and this issues are related to the fine tuning the GA operations. One issue in this case is the fitness scaling and we will discuss about the fitness scaling and their different techniques which are there in the fitness scaling approaches.

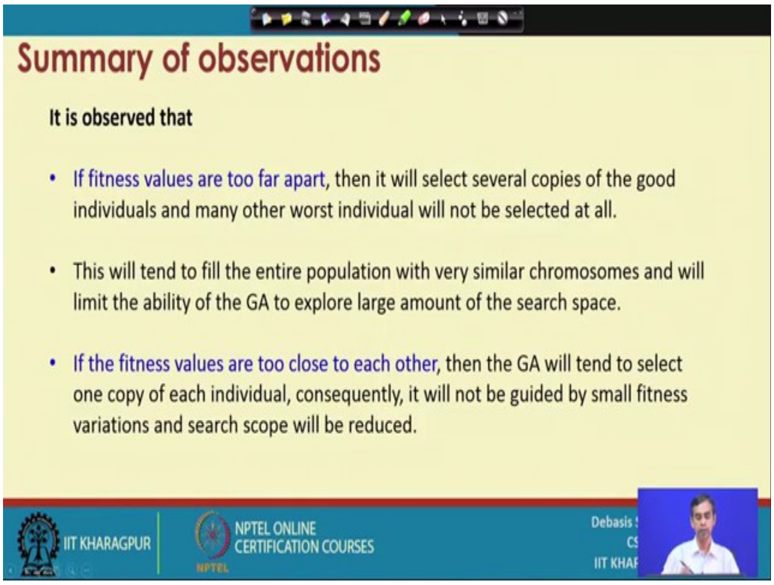
(Refer Slide Time: 25:31)



Now, idea of the fitness scaling obtained from one what is called the situations the situation can be explained like this one, suppose at any instant of the searching. So, these are the solution it is available. Now if we check the range of the fitness values of the solutions. So, we see that these are the range; that means, the lowest fitness value to the highest fitness value this range.

Now, sometime this range matters a lot. So, this range in fact, signify whether there will be premature convergence or inaccurate result and everything. Now let us consider few situations how this gap between the lowest to highest fitness value matters; that means, if the high gap how it works if the gap is narrow then how it works or whatever the gaps it is require.

(Refer Slide Time: 26:27)

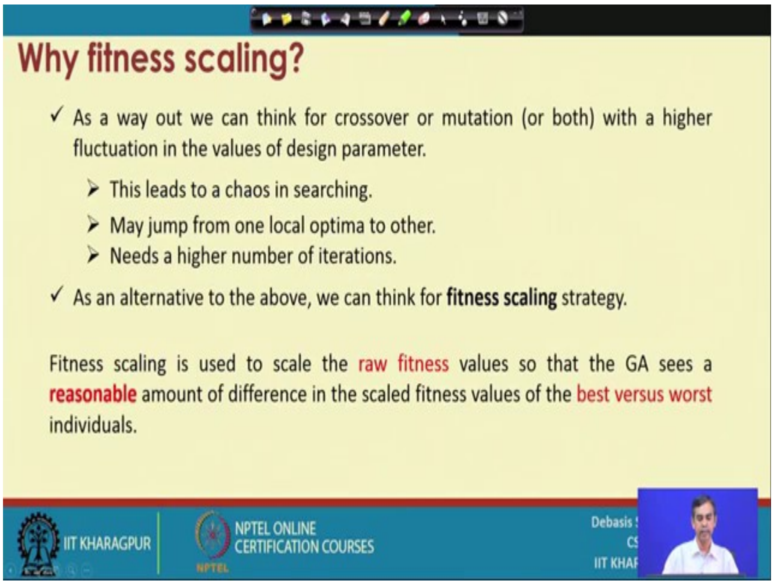


So, it is basically two trade off cities there if fitness values are too far apart mean; that means, they are having very wide gap then it will select several copies of the good individuals always. So, all is and many other worst individuals may not be selected at all. So, this is the one issues are there.

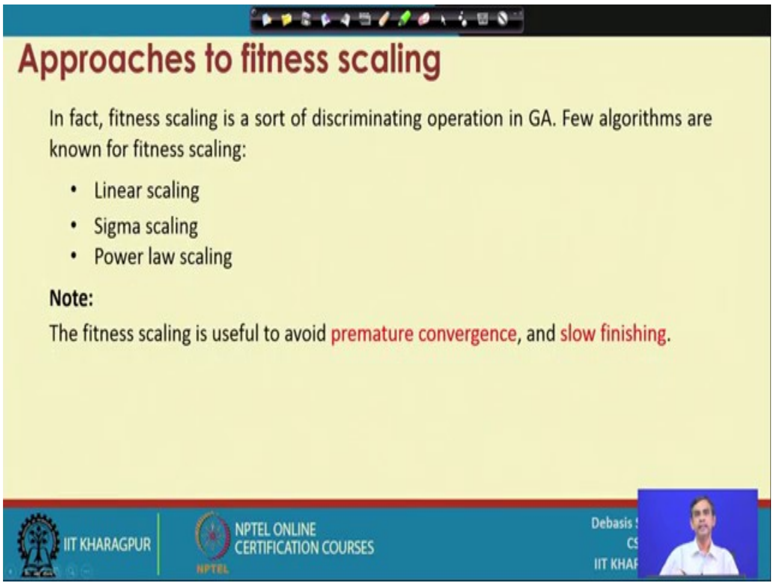
So, this will basically tend to fill the entire population with very similar chromosomes, and eventually it will terminate to a local optimum possibly. On the other hand, if the fitness values are too close to each other; that means, the gap is very narrow, then the GA will tend to select one copy of individual and conjugately it will not be guided by the small finance variation, and such scope will be terminated.

So, it is basically now reduce such scope will be explored. So, both the techniques both the consequence situations have their own limitation. So, this means that we should have the fitness values of the individuals in such a way that the gap between the highest to lowest should not be narrow neither or it should not be the wider again. So, there some reasonable gap how this reasonable gap can be ensured, we will discuss and the different techniques are there.

(Refer Slide Time: 27:40)

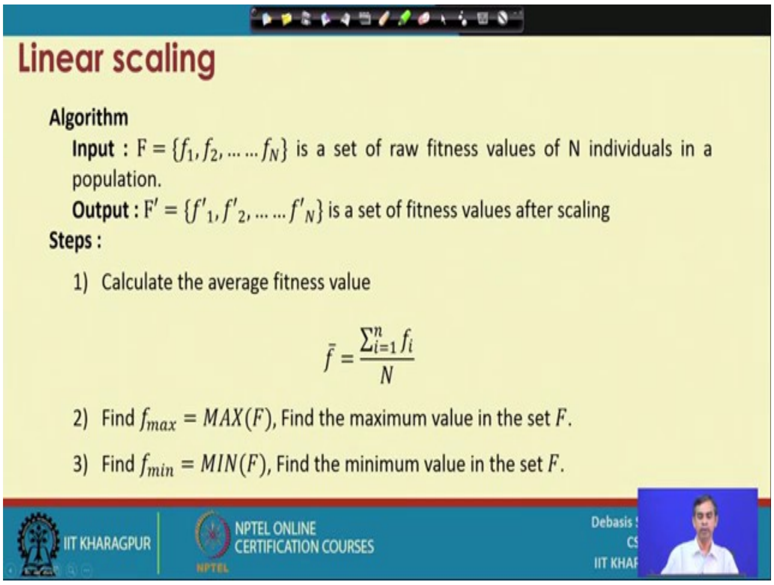


(Refer Slide Time: 27:44)



So, basically idea is that from the raw fitness value we have to evaluate the better fitness value, and there are three techniques usually followed it is basically linear scaling, sigma scaling and power law scaling.

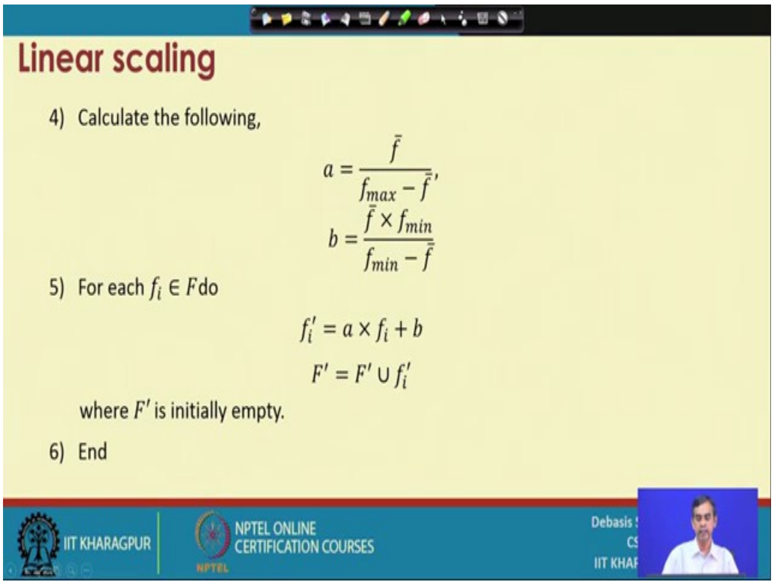
(Refer Slide Time: 27:56)



We will quickly cover the three techniques here. The idea of the linear scaling is discussed first. So, here basically these are the raw of fitness values of the current population where the n number of solutions are there, and this algorithm linear scaling will produce the scaled fitness values. That means, fitness value should be changed so that the gap between the lowest and highest is reasonable.

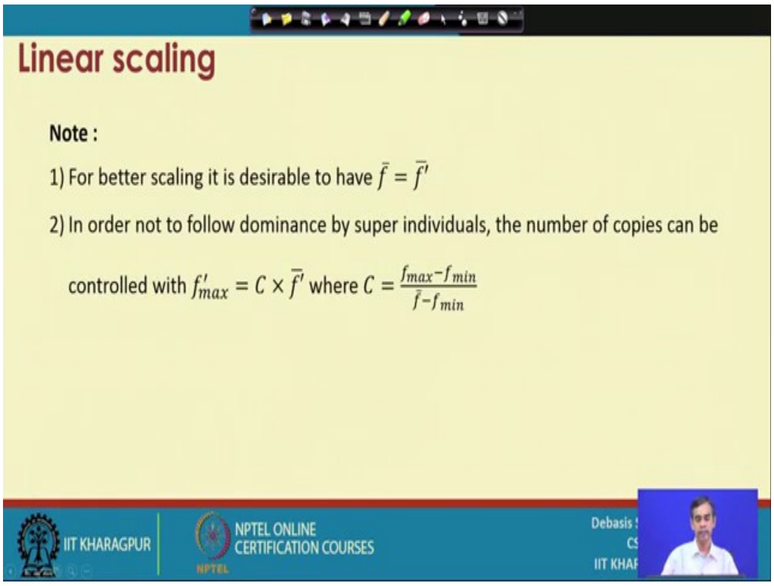
So, idea it is there. So, in this process we have to calculate the average fitness value using this formula. So, it is basically average of all fitness values and then it calculates f_{max} and f_{min}. That means, that has the highest fitness value and this is the lowest fitness values one these two values are obtained then it basically follows the decision how to change it.

(Refer Slide Time: 28:39)



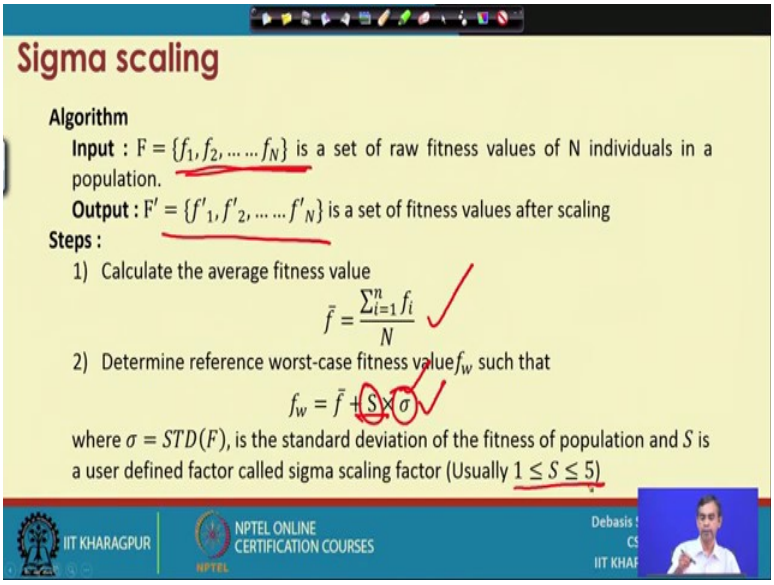
So, basically in this method, linear scaling it computes $a \wedge b$ the two value using this formula. Once this two values $a \wedge b$ known, we will be able to obtain the scale fitness value using this formula, and this is the fitness value it needs to be added into the F' where F' is basically, the set of all the scale fitness value and F' is initially empty. So, this is a method that is followed there in case of linear scaling.

(Refer Slide Time: 20:07)



And then there is another idea about this is called the sigma scaling another technique. In case of sigma scaling, input and output are the same as the previous one.

(Refer Slide Time: 29:11)

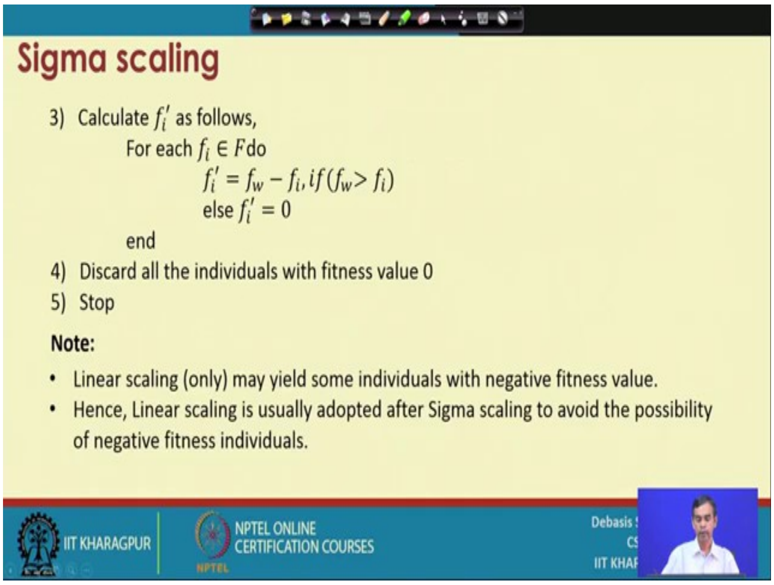


It calculates the average fitness value it is there. But the different between the previous linear scaling and sigma scaling is rise here. So, in this method we have to decide two parameters, the S and σ where the σ is basically the standard deviation of all the fitness value it is there.

So, it is basically standard deviation and S is the one factor. It is also called the sigma scaling factor and usually this value is in between 1 and 5. It is a standard procedure that is followed value. The lowest value of the S as lowest as small as 1 and then highest value is as 5.

So, one these values are known to us we will be able to calculate f_w one calculation for the entire population. So, f_w is calculated based on this formula.

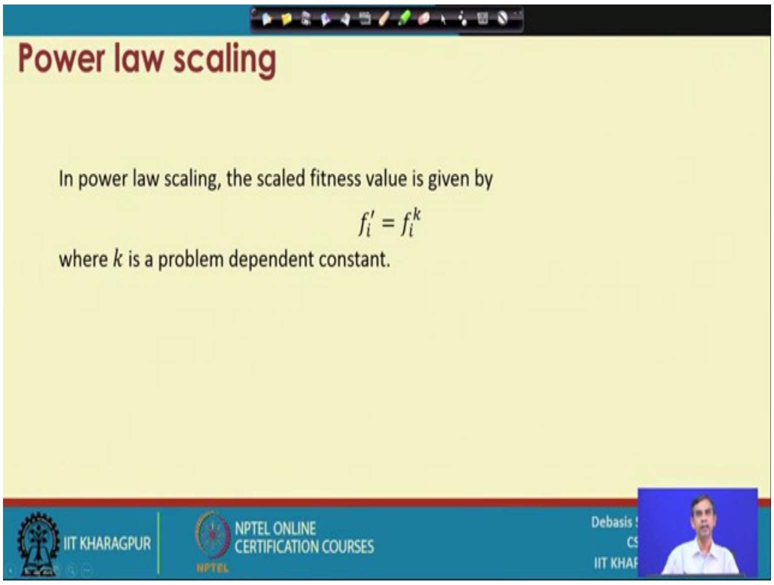
(Refer Slide Time: 30:07)



Once this f w is known we can use this f w to calculate the raw fitness and the scale fitness value. So, for each $f_i \in F$ that is a given population, we have to calculate the scale fitness value $f_1^{'}$ = $f_w - f_i$. If $f_w > f_i$ and it is 0 in if this is 0.

So, this way the scale fitness value for the entire population can be obtained. So, this concept is followed there in case of sigma scaling, and usually the sigma scaling is followed the linear is followed by the linear scaling because some time linear scaling can result some raw scale fitness value is a negative which is not acceptable. So, we can follow the sigma scaling after the linear scaling so that the more refined scaled feature fitness value can be obtained.

(Refer Slide Time: 30:59)



So, this is the idea about the sigma scaling and the simplest scaling is called the power loss scaling. It is very simple idea is a naive approach we can say, if f_i is the current fitness value. User has to decide a k value, k it usually some constant when including the real value also 1.5, 1.2, 2.5 or 2 whatever it is there means how much you have to have a variation.

So, there then the fitness scale fitness value can be obtained by means of this exponential calculation and then this is the idea. So, this is very simple and straightforward method. Sometimes it is we followed there in order to have a very good gap between the lowest and highest value. So, this is the method.

So, we have disused about the different scaling operations and the fitness scaling basically and. So, this includes the operation that is there. So, far the GA reproduction is concerned, we have learnt about the GA reproduction which includes a crossover then mutation and then scaling operation convergence criteria. And we will discuss about the next new topics in the next class it is basically multi objective optimisation.

Thank you.

Introduction to Soft Computing

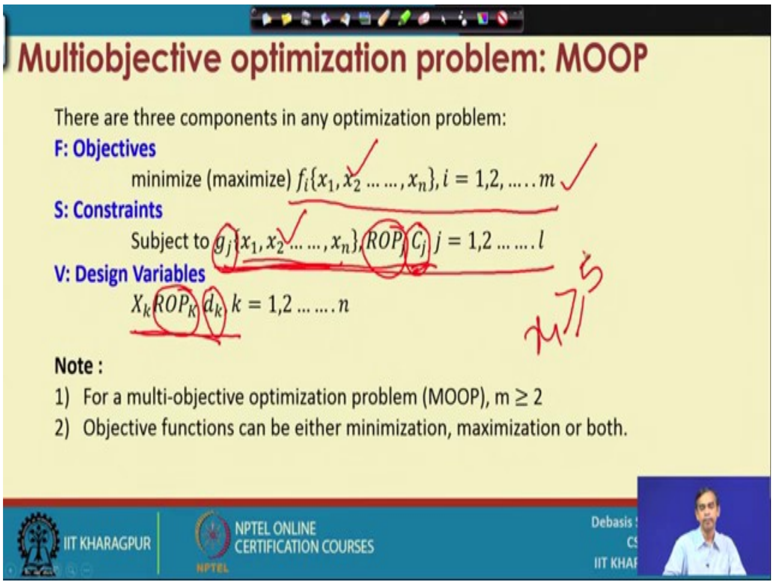
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Science, Kharagpur

Lecture – 24
Multi-objective optimization problem solving

We are discussing about solving optimization problem and we have discussed one specific problem and used to solve that specific problem the genetic algorithm that we have used. Now, the specific problem in the sense that we use we consider only one objective function to be optimised. So, one objective function subject to a number of constants when a number of design parameters are involved. So, this is a special case.

Now, we are going to discuss more general cases of solving optimisation problem where instead of only one objective function there will be two or more objective function. So, this particular problem is called multi objective optimization problem.

(Refer Slide Time: 01:16)



So, in a formal notation the multi objective optimization problem we will may abbreviate as a MOOP, the short form of multi objective optimization problem where we can define in a formal specification which is here. So, like the objective functions related to the single objective optimization we have understood that objectives, constants and design parameters.

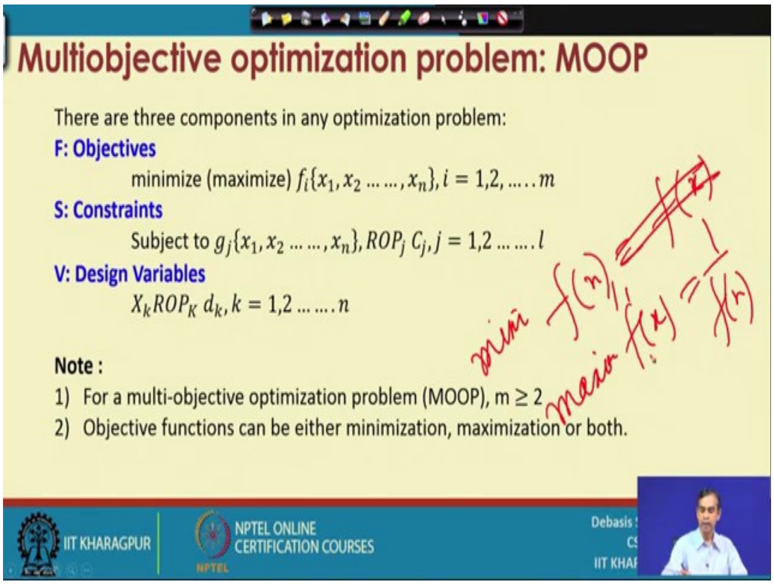
So, here we can see the difference between this definition in the context of multi objective optimization problem. Here the objectives are a number of objectives. In this expression, we have disused about m objectives and out of these m objectives some objectives are to be minimized or some are to be maximized or all objectives are to be minimized or all objectives are to be maximized. And the remaining parts are the same as in case of single objective function. So, there are constants. So, here we have stated l number of constants are involved and those constants are expressed by the function say g_j and you can say that all the constants are expressed in terms of all the design variables. So, here the design variables which are used to define an i-th objective functions are the same design variables are used to define the j-th constants and these are the design parameters design variables.

Now, here I want to see here. So, all these are the constants are related to some another constant are related to some constant and they are related by this operator. It is called the relational operator. That means, every constant has different relation operator may be say equals to less than or greater than some constant. So, these are the general expression in fact, right and then the design parameters also express in terms of some other relational operators and they are also related with some constant for the k-th parameter for example, say if x_1 is a one parameter $x \geq 5$. So, this is the one constant.

So, these are the different, these are the statements by which we can express the multi objective optimization problem. And as we have mentioned it is there for the multi objective optimization problem this values of m should be at least 2 and the objective functions are there which are to be optimized either on the subject to the minimization problem or maximization problem or both.

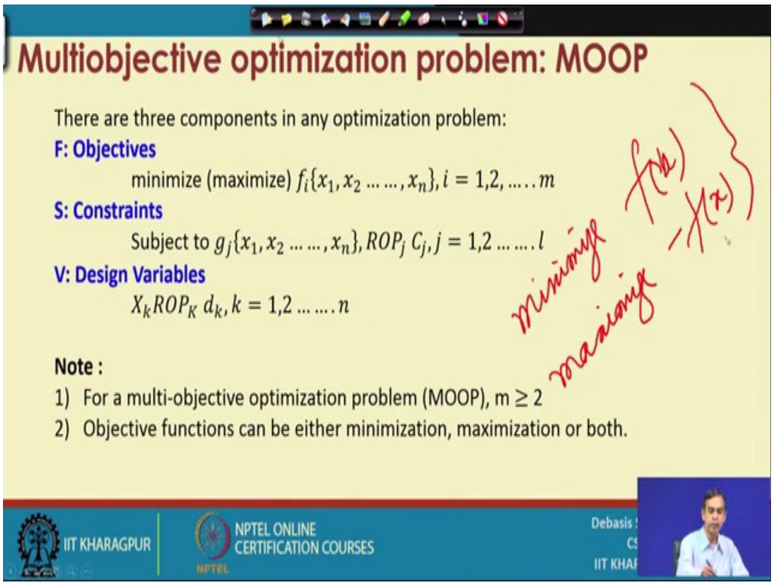
Now, here I want to say one more thing is that in any objective function which is to be minimized can be equivalently converted to a maximization problem and vice versa.

(Refer Slide Time: 04:31)



For example, $f(x)$ this is equals to some function x if this is a problem for minimization then the same objective function, so it suppose minimize. Then the same function can be solved at maximize $f(x)$ or we can say $f'(x)=1$ by; so, it is minimize $f(x)$ when $f'(x)$ is basically $\frac{1}{f(x)}$. So, basically which is the reverse inverse of the function is basically maximization. This is another way, another way also we can write.

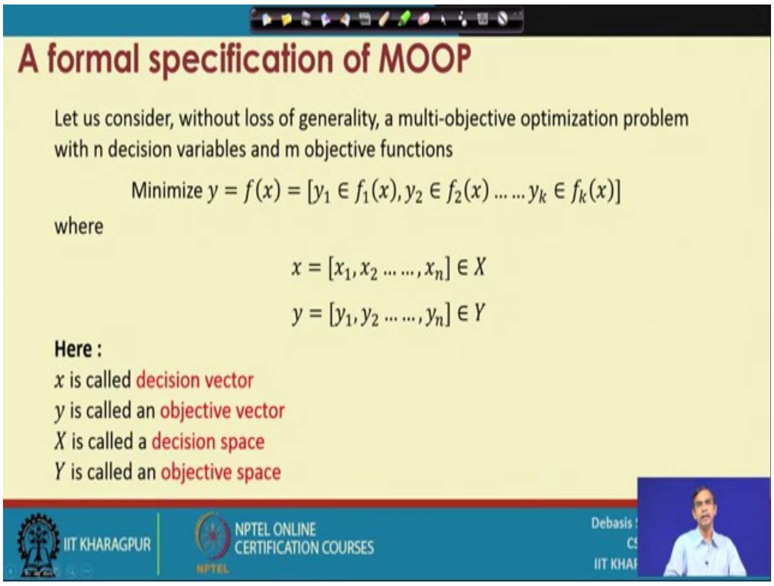
(Refer Slide Time: 05:18)



Say one function is minimized $f(x)$ the equivalent maximization problem can be maximize minus $f(x)$. So, both are the same. So, these are the, I mean transformation from minimization to a maximization problem. What I want to say is that if there are some objective function are minimized or maximized then we can express uniformly either belongs to all functions are minimization problem or maximization problem. Now, the statement by which a minimization problem can be converted to maximization and vice versa is called duality problem. So, we can apply the duality problem to transform the objective function from the form of minimization to the form of maximization.

So this is about the problem that we MOOP statement, statement of MOOP problem and having this understanding, will discuss about what exactly the problem it is here, anyway.

(Refer Slide Time: 06:18)



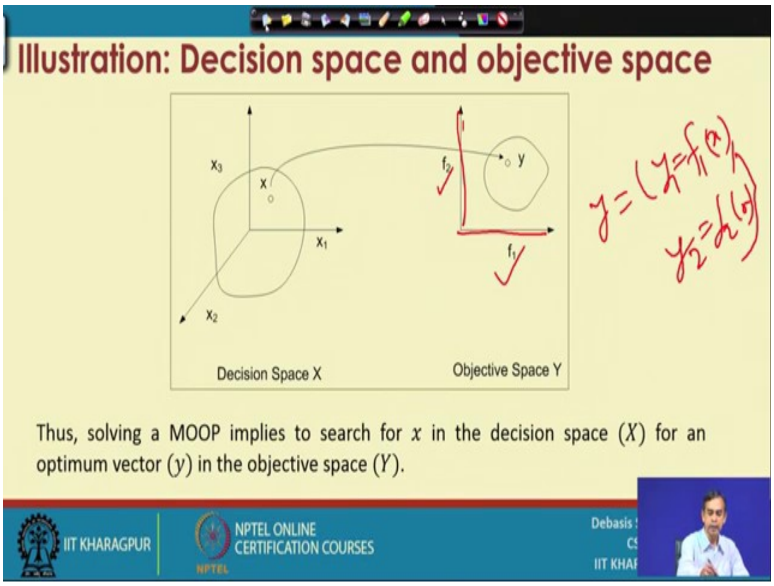
So, in a more mathematical expression we can tell about the MOOP problem where n number of design variables and m number of objective functions are involved and without any loss of generality we assume that all objective function are to be minimized. Then what we can say is that in fact, minimizing all the functions or all the objective function will give some results which is basically minimizing one another what is called the combined function. So, we say that $f(x)$ is a combination of all the objectives that is there and then these are the output that this combined objective function will return. So, here is y is basically an output and all the design variables solution design parameters are the basically input to the problem.

Now, so we can decide about in the context of this expression. So, these are the x , x is basically set of design variables which is denoted here more specifically and all the values of x which are like this they belong to some set it is called x . Similarly, y it will return the output which objective function. So, all these objective values and we can term this objective functions are the results as an output and all the value of whatever it is this it will be in the domain it is called the y . So, x and y are the two domains. We can say it is input domain and it is the output domain. So, for some values of the design variables it will produce these are the output and so, this is there.

Now, so here more precisely this x is called the decision vector and all the y , y_1 and all this thing is called the part of the objective vector. So, this x is decision vector and y which is basically the results obtained from each function constitute a vector and is called the objective vector and this X is called the decision space whereas, Y is called the objective space.

So, basically as I know that, so far the objective function solving and optimization problem is using genetic algorithm is basically a searching problem. Now if this is the statement of the problem multi objective problem, now, let us understand what basically to be searched here.

(Refer Slide Time: 09:08)



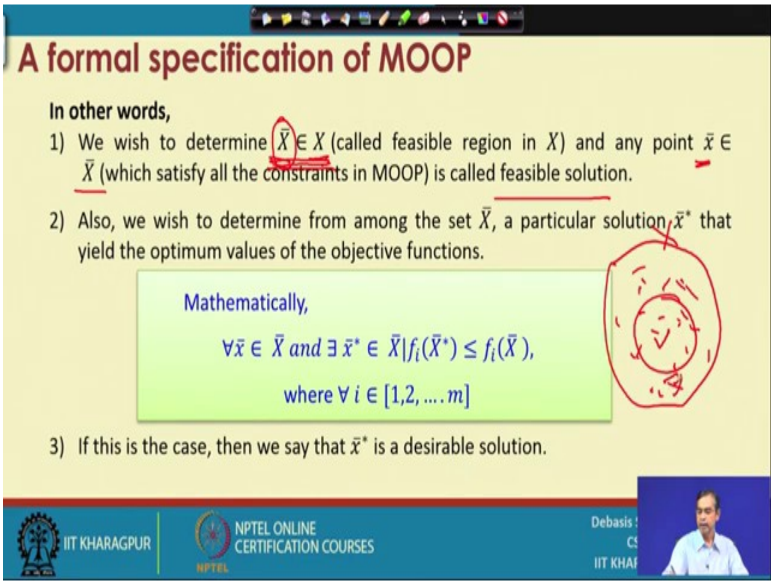
I will explain these things with the only two objective function so that I want to give more graphical impression about the ones concept. Here in this example suppose $f_1 \wedge f_2$ are the two objective functions. So, this y is basically is a vector $y_1 = ¿$ $f_1(x)$ and then $y_2 = ¿$ $f_2(x)$. So, we symbolically represent this concept y using this expression. So, it is basically idea is that for the different values of f_1 and f_2 the different solutions are possible and that is the solution is called this one. So, this is basically an objective vector. So, y is the objective vector here which has the two parts one f_1 part and f_2 part.

Now, again, for input space is concerned that is the decision space. So, this basically represent the decision domain and here we represent a decision vector in terms of two, in terms of three parameters or values that are x_1 , $x_2, \wedge x_3$. So, the same thing can be depicted in the 3 dimensional space and, so for at any instant this point in the decision space represents the instance of a vector having. So, this is the x_1 value, this is the x_2 value and this is the x_3 value. So, it is like this.

So, this way we will be able to express. So, this is the decision space and this is the objective space. Now, for the searching of the solution searching for the solution of an objective problem is concerned is that there is a mapping for every decision vector to an objective vector. So, is mapping is there. So, if we consider another decision vector and then there will be another objective function it is like this. So, there is a mapping from there is a mapping from this phase to this phase and out of this mapping we have to select the best mapping. Best mapping in the sense that the objective this objective vector will give the optimum value.

So, this problem essentially is basically mapping from decision space to an objective space and that is a searching the procedure or the searching policy should be to find the best map.

(Refer Slide Time: 11:41)



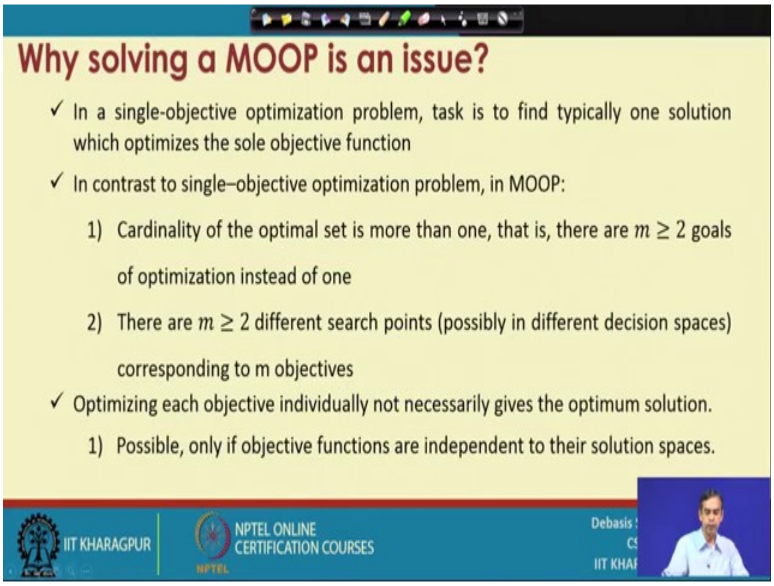
Now, more mathematical again we can define it so that we can discuss the next discussion more in an understanding manner. So, basically the idea is that the X the decision space as we have mentioned earlier is basically the solution region and out of the solution region there is a subset we say that it is $\acute{X}$. So, X is basically the X is basically the solution space where are $\acute{X}$ is a subset of region because all solutions may not be the feasible solution. So, we have to consider the solution $\acute{X}$. So, $\acute{X}$ is, $\acute{X} \in X$. Now, if we consider this $\acute{X}$ and any point say this one $\acute{X}$ into this feasible solution region which basically satisfies all the constant that is there in the MOOP specification then is called a feasible solution.

So, X constitute of many solutions. So, it is basically is the in the domain of decision space all these are the solutions all the possible values, but we have to consider only few things which basically and if this is the X and this is $\acute{X}$ then this basically called the feasible region and the feasible solution. So, it is the idea is that we have to find all the values which are feasible solution first. And out of these feasible solutions we have to select a particular solution and this particular solution we denote as $\acute{X}^{¿}$. So, $\acute{X}^{¿}$ satisfy some constant which is mentioned here. So, again we can explain it like this $\forall \acute{x} \in \acute{X}$; that means, for all feasible solution in the region of feasible solution and there exist a particular solution which we denoted at $\acute{X}^{¿} \in \acute{X}$ such that such that $f_i(X_i)$ bar should be

$\leq f_i(\acute{X})$ if we consider the minimization case, if we consider maximization case then this will be this one.

That means for the values of these $\acute{X}$ and for the values of $\acute{X}$, $\acute{X}^{¿}$. So, this objective functions should be always less than any values in the solution space. So, if the solution space contains the m number of solutions. So, here $\acute{X}^{¿}$ is called an optimum solution or it is basically a desirable solution. So, mapping therefore, comes into this picture. So, from the set of all design parameters values we basically obtain the solution space. From the solution space we have to decide the feasible solution and out of all the feasible solution. We have to select one solution which is called the optimum solution or desirable solution. That means, we have to search the design space for which values X will be a feasible solution and from all the feasible solution we have to search a value which will give us a desirable solution. So, this is the concept that is there in case of multi objective optimization problem solving.

(Refer Slide Time: 15:10)

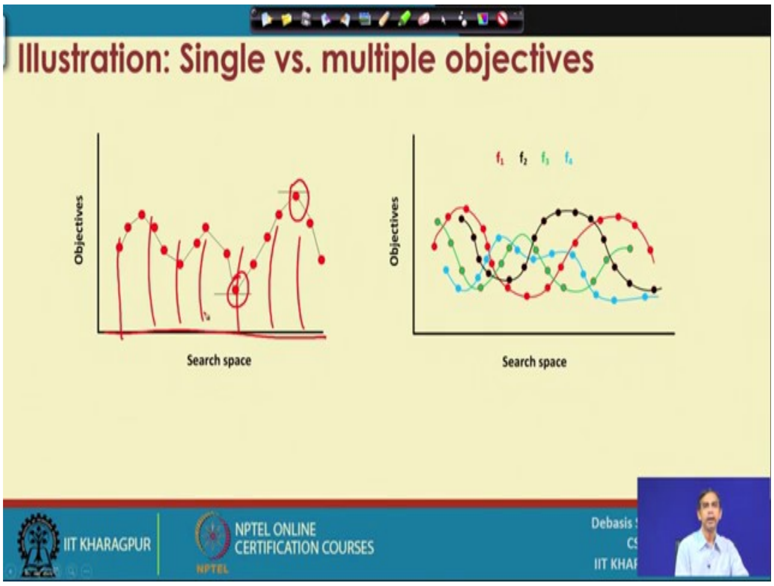


Now, will discuss about few things are involved now the first question that is here. So, if we know the single objective optimization problem solving using genetic algorithm then why we should consider the different what is called the procedure or different techniques or different principles to solve the MOOP problem.

Now, there are definitely many differences between the single objective optimisation problem and multi objective optimisation problem. So, here in this slides I have listed few differences. Now, the first in a single objective optimisation problem that we have learned so far, there the task is to find typically one solution which optimize the only one objective function. And in contrast in case of MOOP problem our objective is to not only a single objective function rather two or more objective function to be optimized. And when there are two or more objective functions this means that it least to two or more search points in contrast to the single objective function where only one search point. So, they are because the different search point related to the different objective function. So, these are the basically constitute and objective vectors or output vectors we can say and then we have to select out of many search points we have to select one search point.

And then optimizing each objective individually it is no issue; however, optimizing the entire what is called the considering all the objective function putting together and then finding a global optimisation is in fact, a non trivial task. Now, we can explain how this become a non trivial task that can, that can be explain with an example.

(Refer Slide Time: 17:06)



Here if we see the two graphs in the left side of the graphs we show the search space for a single objective function and it is as usual, so it is basically you if you have to find a minimum solution then it is the point that can be searched to find it or if it is a maximization problem then this point can be search to find it.

So, the single objective function is concerned is only one objective function this is the search space and for each search space we can find all the solutions and they ultimately find either maximum global maximum or global minimum value. So, this is the simple problem and we have learned about how GA can be applied to search the entire search space.

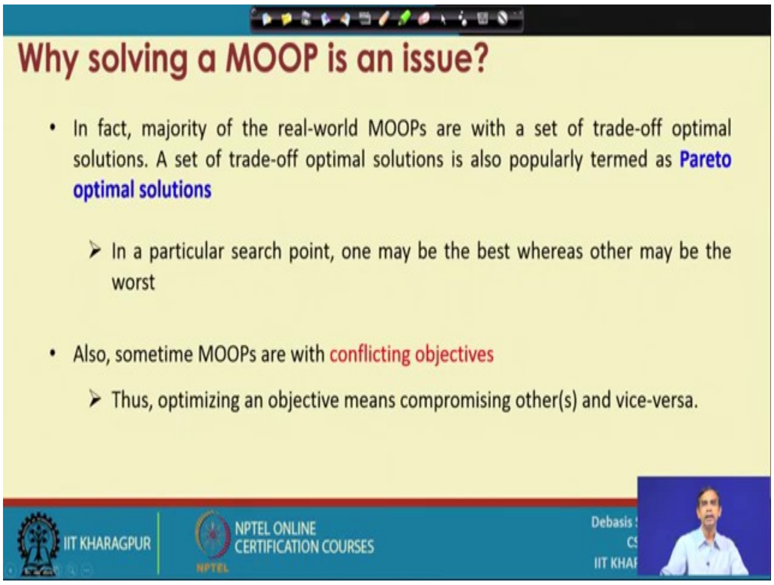
Now, on the other hand in case of multiple objective function in the right part of the graph if we see this graphic little bit carefully we see that we have plotted the objective function value with respect to the search space. And the different curves, with different colours basically represents a different function like f_1 , $f_2, f_3, \wedge f_4$, and if we see it then it is easily visible it is easily understandable that the different objective function has their different points for which the minimum and maximum occurs. For example, in case of function f_1 if we see the minimum, this is the minimum value. On the other hand, if we consider the function f_2 then the minimum value is here. So, this point if this one is the solution space than this one is the solution space there.

Now, again for f_3 if we consider this minimum value maybe it is here. So, solutions space the and similarly for f_4 we can find the solution space here, so this one. Now, what we have conclude from there is that for the different objective function if it is taken into consideration then their solution space will be anywhere in this region; that means, in case of single objective function if only one search point is there, but here the number of search points are there. All search points are related to search point is related to, the optimum value with respect to a particular objective function.

Now, here the question is that out of these search points which point should be taken in order to objective I mean consider the multiple objective function optimisation. So, definitely it is neither this solution or this solution or this solution we have to select out of these one only one solutions in fact. Now, which solution needs to be selected and how these solutions can be explored this is the issue. One issue is that single objective is very fast to do it, but as this number of objective function is it is obviously, the multiple cause to be involved in order to sub optimize this one.

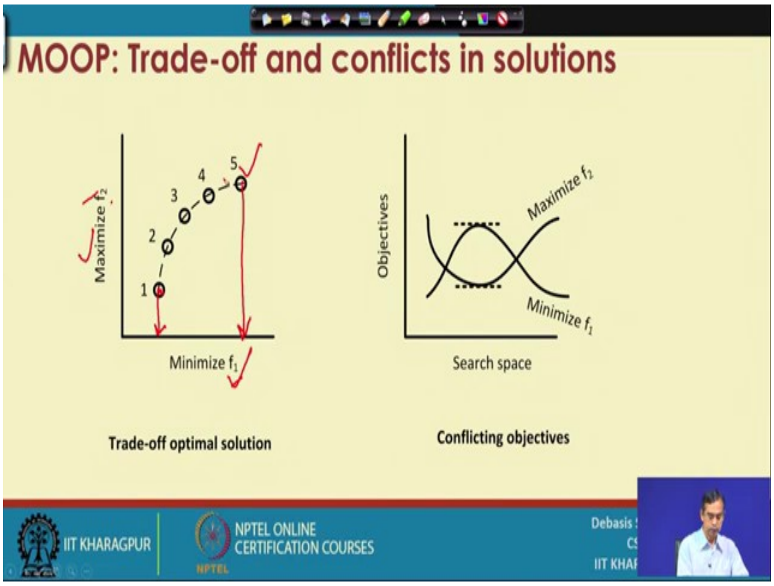
So, the traditional genetic algorithm that we have learned is not sufficient to solve this objective function in fact so we have to study a totally different concept to solve the MOOP problems, multiple objective optimisation problems.

(Refer Slide Time: 20:15)



And this is because there are some objectives which are conflicting in nature because if we select this one solution then other with respect to other objective function this may not be solution. I can give an example to understand this concept more clearly.

(Refer Slide Time: 20:39)

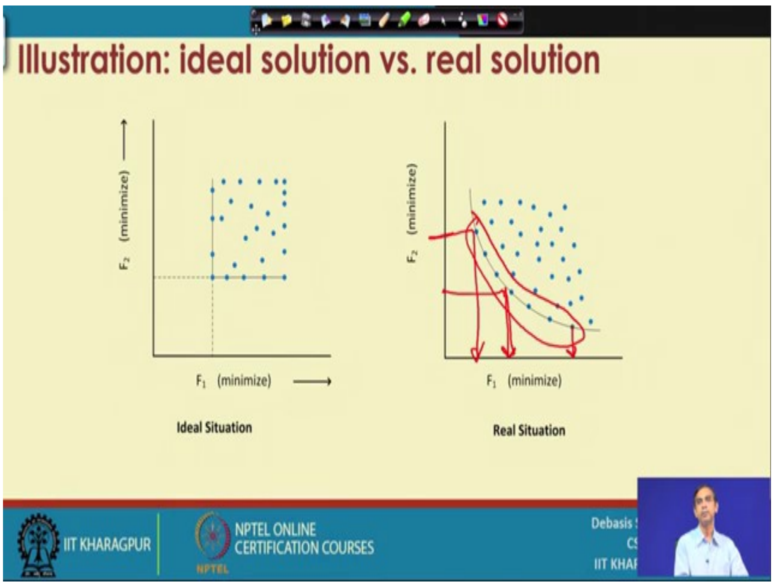


So, here one example I can see. So, these are the again with respect to two objective function f_1 and f_2 and how the different solutions are there in order to solve the values it is there. So, there is a one solution, 2, 3, 4, 5 these are the solutions.

Now, we can see if we consider this is the solution one and it is a minimization problem then definitely this solution is preferable both with respect to f_1 and f_2. Now, so this is a maximization then means that this solution the solution one is preferable with respect to f_1. However, this solution is not preferable with respect to f_2. On the other hand, if we consider this solution this is preferable with respect to f_2 because it gives a maximum value for f_2. However, it gives the worst value for f_1. That means so it has one what is called the point is good, but other point is bad similarly this one, and here 2,3,4 either neither f_1 is good nor f_2 is good.

So, this is the concept which called the conflicting objective function. So, objective functions are conflicting f_1 conflicts with f_2 and vice versa. So, the same things which is mentioned here it is shown here. So, these are the fine; if we consider this is the search space then with respect to the search space this is satisfiable, but this is not acceptable. Now, it is like this. So, usually in case of multi objective optimisation problem solving the objectives are conflicting in nature and therefore, finding a unique solution out of these different values of the objective function is a tedious job.

(Refer Slide Time: 22:34)



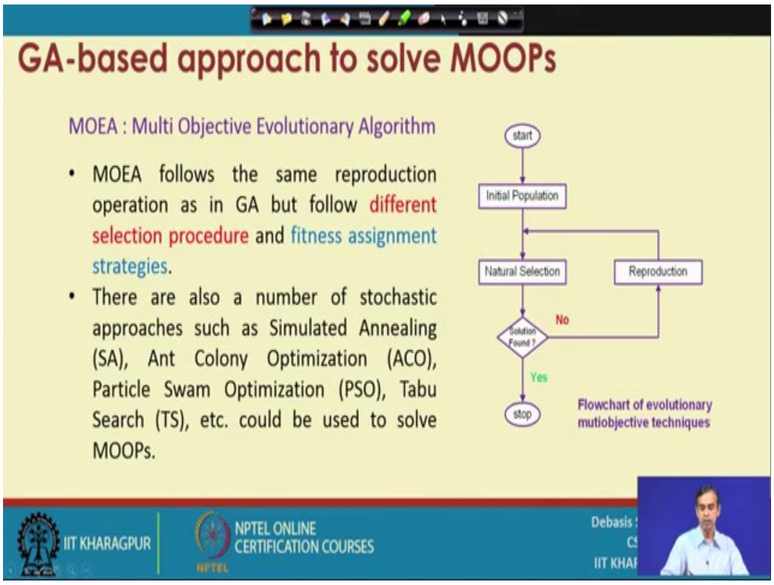
Now, another example, usually there may be some situations if we are very much fortunate enough then we can find a unique solution to solve where both objective functions are satisfying without any conflict.

For example, here, suppose F_1, F_2 both are to be minimized. Now, if it is a minimization problem with respect to all the objectives and these are the solutions suppose. So, these are the solutions and out of which so many solutions we can see one particular solution which is comparable to any other solution in the solution space. Now, this solution satisfies or it is basically desirable solution because it minimises both F_1 and as well as F_2. So, in this case this is a solution which is non conflicting. Now, this is a solution is called ideal solution and the situation where we can get this kind of scenario is called the ideal situation.

So, usually ideal situation is very very far from the real solution. In fact, it is observed that most of the objective functions in case of multiple objective problem solving the solution is like this kind of things are there. So, here basically these are the solution region solution space and out of the solution space we can find some solution it is this kind of solution actually which is basically neither superior to anyone. For example, if we consider this one and then minimization if we both F_1 and F_2 then it is good with respect to F_1, but not both F_2. Similarly, it is not so good with; it may be good with respect to F_1, but not good. Similarly this is one is good with respect to F_2, but not with respect to F_1.

But so, what I want to say is that all the solution that is here in this line there the boundary you can see they are any one solutions which lies on this boundary are neither superior to any one such a solution in multiple objective optimization problem has the special importance. In fact, what is the special importance is that if this is the solutions we can find from the search space then we have to take or we have to select one solution from there for all solutions are acceptable although it is not ideal solution. So, such a solution particularly in the theory of MOOP optimization problem is called the pareto optimal solution means a pareto solution. Means all the solutions are we have to consider in order to decide your own solution. We will discuss about the concept of pareto solution and then pareto optimum solution not in this lecture in other lectures in due time.

(Refer Slide Time: 25:19)

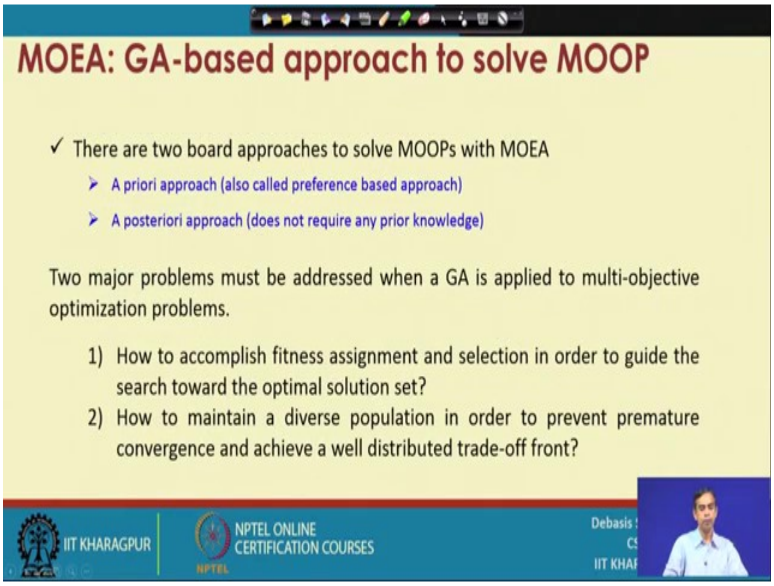


Now, let us see if we want to solve the genetic algorithm basically our objective is basically to apply genetic algorithm to solve multiple objective function, multiple objective optimisation problem. So, idea or the frame work that is used is basically same as the frame work we have considered to solve single objective function that is a one important point and is a, I mean good point to learn it so that the same framework can be applied. However, in the framework we have to do little bit different tactics or the different techniques to be followed, anyway.

So, the idea the basic task that is there in case of genetic algorithm is also followed here namely initial population creation and then selection, and then reproduction, and this is the loop between the selection and then reproduction producing the next generation and so on so on, but the different is there. Different is that whatever the method that we have discussed for the selection that is there in case of single objective optimization is not allowed or not applicable. Mainly the selection operations are to be fine tuned, so for the solving multiple objective optimization problem is concerned using genetic algorithm.

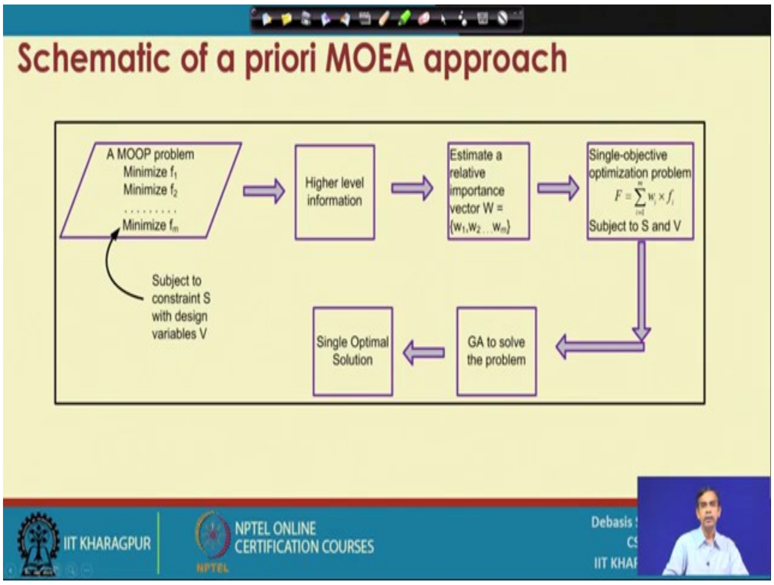
So, solving MOOP problem using genetic algorithm basically the idea is that how the different tactics or procedure that can be used to using the selection procedure and therefore, the fitness assignment concept there because it is basically to assign the fitness so that we can lead to a better search there. So, these are the thing that we learn it. Now, I am telling you again, let me clear one more thing is that. So, GA is the one approach, parallel to GA there are many other approach in the line also to I mean help us to solve multiple objective optimization problems like Simulated Annealing, Ant Colony Optimisation problem, Particle Swam Optimization problem, Tabu Search and like this. So, all this thing are the many many theory many techniques many concepts are there, but here will limit our discussion to the genetic algorithm based approach.

(Refer Slide Time: 27:51)



Now, so for the genetic algorithm based approach is concerned there are may be two broad techniques, one is called the a priori approach and another is called the posterior approach. Now, will discuss about the two techniques in brief and then what is the procedure that is followed there.

(Refer Slide Time: 28:11)

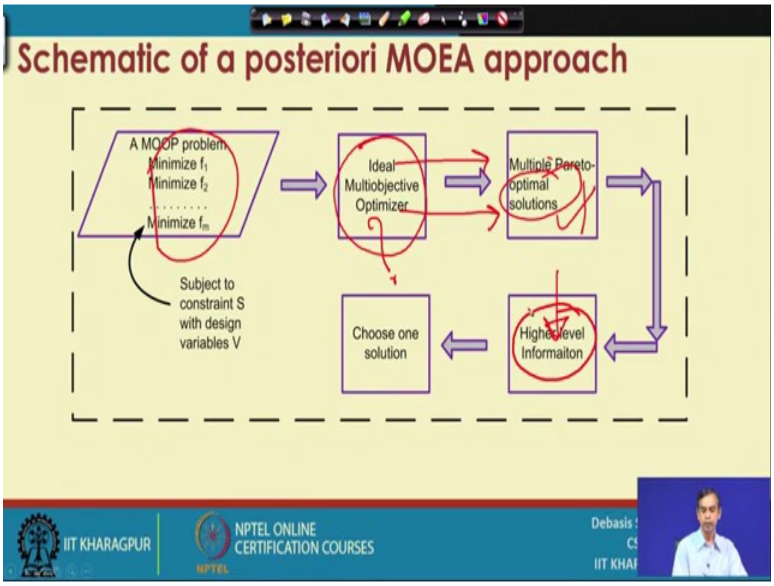


Now, here is basically the flow of the a priori approach, it is easy to understand, so this basically the problem your MOOP problem where m number of objectives are to be optimized. Now, in this approach it is called the a priori approach because it follows certain high level information. Now, what is exactly the high level information I will discuss this lecture within a one minutes or so. Now, we will understand about high level information.

Now, this information basically helps you to understand one weight vector it is there; that means, what is the weightage of the different objective function to be considered in order to decide one solution of our own from the set of solutions provided by each objective function. For example, if $w_1 = w_2 = w_m = 1$, then all objective functions are equally important, but if w_1 is highly important then w m then I can give more weight age to w_1 then w_m . So, these are the, this one. Now, in order to have the weight values we have to follow the high level information.

Then in terms of this weights we can express the multiple objective function as a single object function is basically summated weight form $w_1 f_1 + w_2 f_2 + \ldots + w_m f_m$. So, this basically gives the m. So, here basically what is our tricks is that considering the multiple objective optimisation problem and transforming this into a single objective optimisation problem in terms of the weight vectors. Then if it is a single objective function then there is no issue we can apply our traditional genetic algorithm on this and then find the solution. So, this is the aprior approach and then we will discuss about another approach it is called the posterior approach.

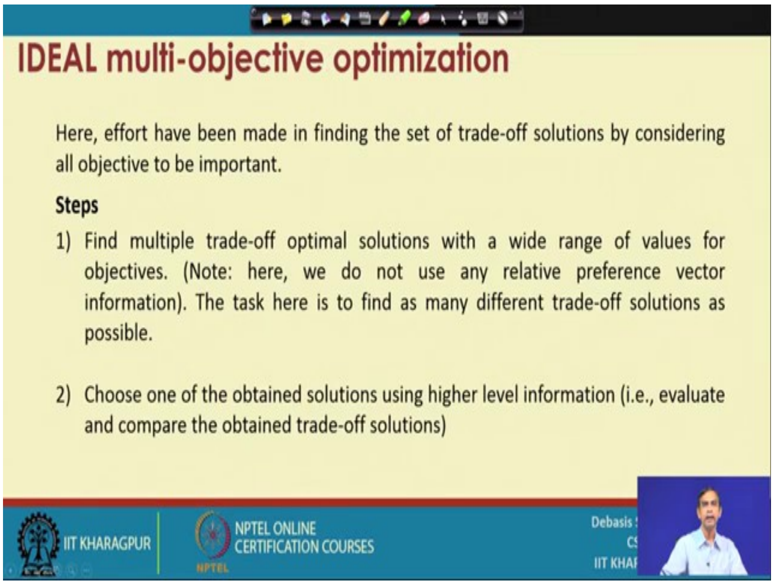
(Refer Slide Time: 30:04)



That means we need the information, but at a later stage it is as usual in case of objective function, this is the problem statement, this is basically ideal multi objective optimisation problem, this is a new thing we have to think about it. That means, it will give you the way how the multi objective optimisation problem can be solved.

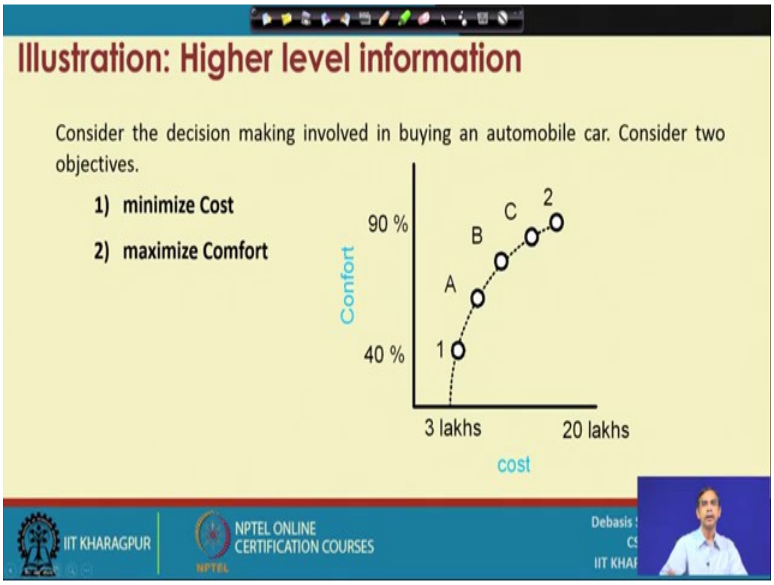
Now, if we solve it right, for example, one simple idea about ideal multi objective you just solve one objective function at a time. So, this means that it will give you a number of what is called the solutions as we have learned about it. So, this, the solution is called the pareto solutions or pareto optimum solutions. Then we use this and pareto solutions and passes through one high level information because we have to see this will give you a large number of solutions and we have to select only one solution from there, and then the high level information can be used to select a particular solution that is a desirable solution or our required solution.

(Refer Slide Time: 31:05)



So, these are the two approach that is there and now I will discuss about idea about, so is basically high level information.

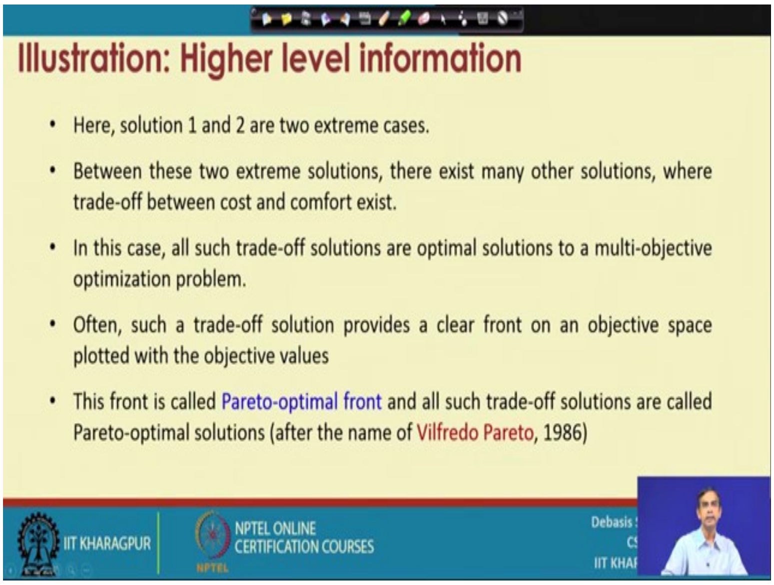
(Refer Slide Time: 31:10)



I can discuss the concept of high level information with an example. Say suppose you have to purchase a car and there are two objectives of course, the cost and then comfort. That means, you have to purchase a car with a minimum what is called the cost involved and then maximum comfort possible from the car. Now, if you visit many cars those are there in the car markets then you can find a number of solutions there, for every car have their own cost as well own comfort.

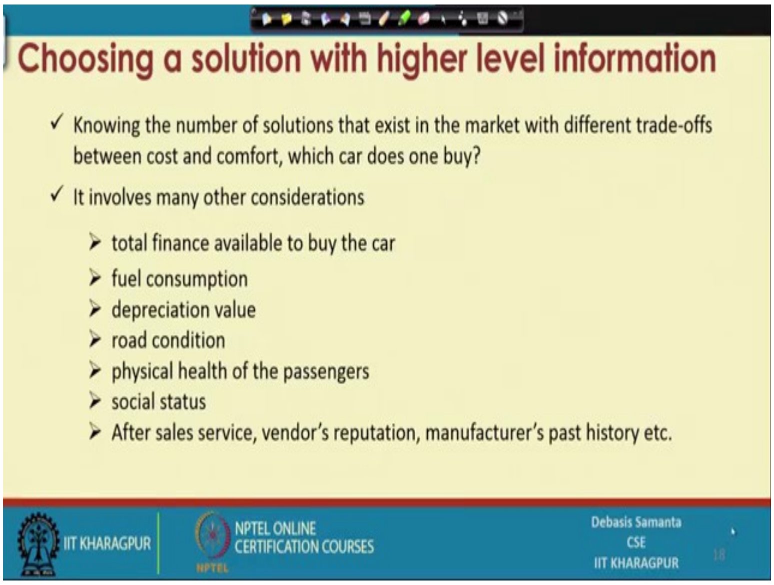
So, here we have given few examples. So, five solutions suppose we have served and then these are the solution like this one out of which this car is good so for the cost is concerned, but not for the comfort, but this car is so for the comfort it is concerned it is preferable not for, so for that cost is concerned. So, then we have to find out of this the solution is there. Now, we can follow the high level information there.

(Refer Slide Time: 32:10)



Now, I will discuss about what are high level solution are there.

(Refer Slide Time: 32:14)



So, they are high level solution right in the context of this car purchase, what are the total finance that you can give to buy the car or what is the fuel consumption each car, depreciation value and in which road condition which car travels better and then physical health of the passengers if a particular car is used and then social status and all these things. So, these are the different high level information if you take into consideration above the solution that you have obtained then it will help you to decide the right solutions or desirable solution.

So, high level solution regarding this thing we learned a lot whenever we discuss more theory about it. So, this is the concept of multiple objective function that we have covered in today's class. In the next class we will discuss many theory and then some treatment and then how to solve these problems in a more pragmatic way, will discuss in the next class.

Thank you very much.

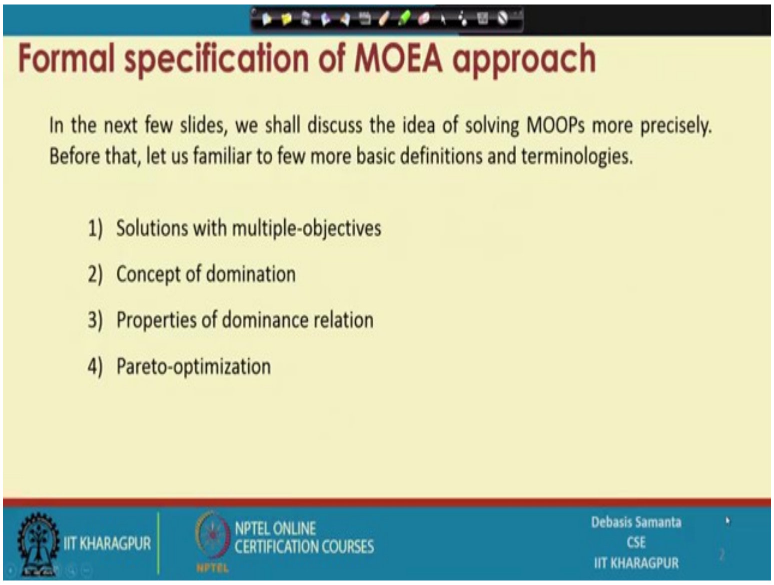
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Science, Kharagpur

Lecture – 25
Multi- objective optimization problem solving (Cont.)

So, we are discussing about solving optimisation problem. In the last few lectures, we have learned about how to solve single objective optimisation problem. In the last lecture, we were discussing about solving multi objective optimisation problem which are more applicable in our real life applications.

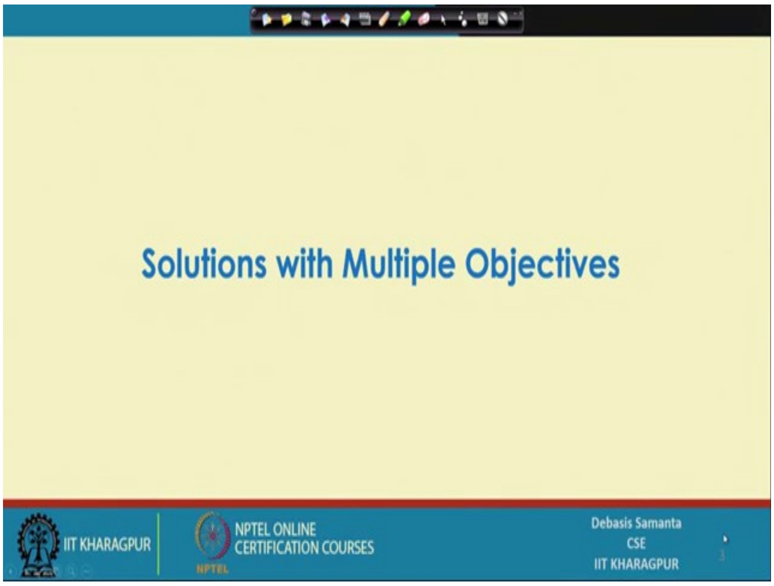
Today, we will continue the same discussion today mainly we will discuss about some properties, some characteristics which is very much essential to solve multi objective optimisation problem. And after we learn the different characteristics then we will discuss the different approaches to solve multi objective optimization problem.

(Refer Slide Time: 01:06)



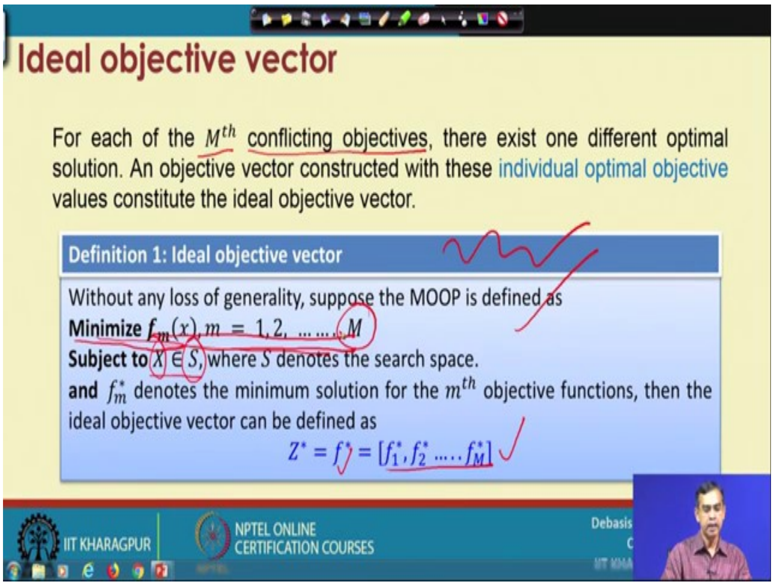
Now, so we have planned our discussion in subsequent lectures like. First, how the solutions with multiple-objectives are characterized and then the solution possesses some important concept it is called the concept of domination we will discuss it. And then we will try to learn about the relation that is the properties that a dominance relation holds. And finally, we learn about Pareto-optimization techniques or use a Pareto-optimization concept. So, all these things we will done one by one.

(Refer Slide Time: 01:45)



So, let us first start with solutions with multiple-objectives, and how they can be characterized.

(Refer Slide Time: 01:54)



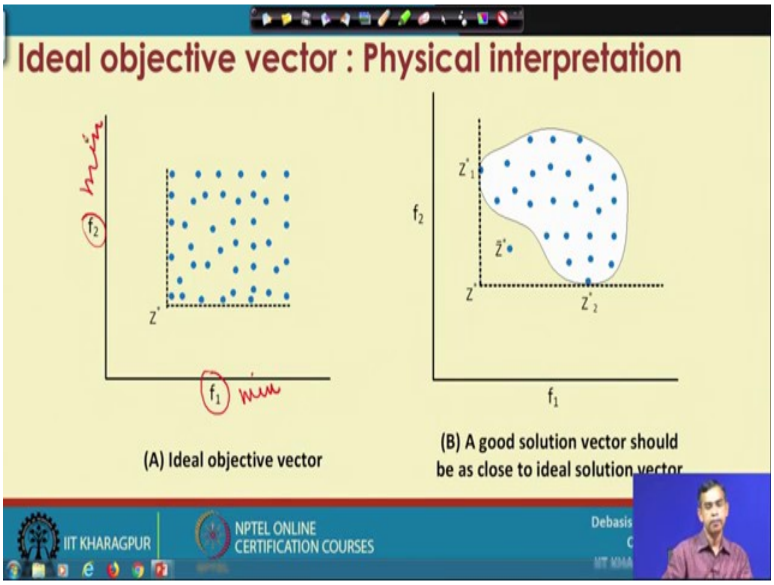
Now, let us consider for our discussion say that there are M number of objective functions. And these objective functions are conflicting objectives. Conflicting objective means if you try to minimise both say f_1 and f_2 then when you try to minimise f_1 , f_2 not necessarily to be minimised or vice versa that means, there is a trade off. If we want to minimize both then there may not be any solution. So, we have to have some conflicting objectives if it is there then essentially there exist many optimum solution.

Now, if it is possible to exist one different solution which is basically satisfying both the objectives simultaneously, then we can say that such a solution is called the ideal solution. So, solution which has the ideal solution also called solution ideal solution and then the objective functions which basically holds good like this it is called the ideal objective vector.

Now, so here the same thing is discussed in a more precise or more formal method formal way. So, say here basically for the simplicity, we assume that all the functions all the objectives are to be minimized. So, it is basically it is generalization, it can be there are may be some objective function to be minimize some to be optimized, but we know with the virtue of principle of duality problem, all objective function can be converted to only one type. So, let us consider there are M number of total objective functions and out of which each objective function are to be minimized.

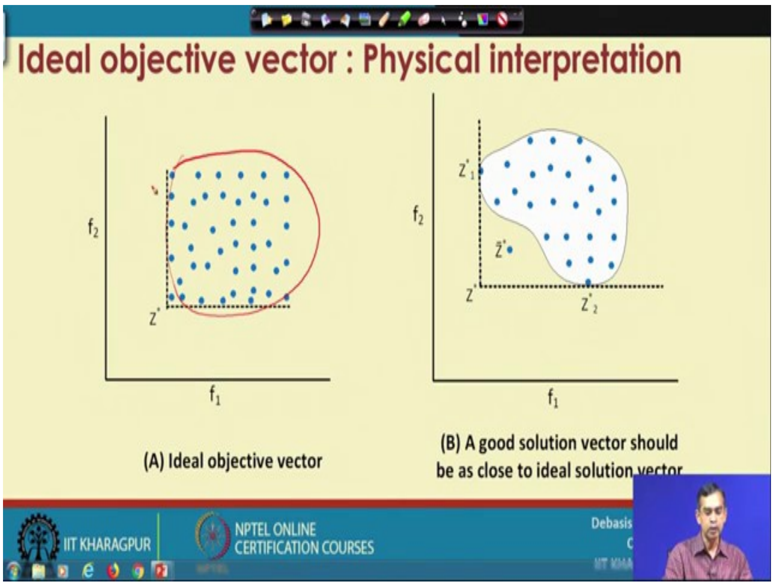
And then they are basically subject to $X \in S$, where S denotes the search space that mean we have to find a solution X in the search space S which basically satisfy all the criteria simultaneously. That means, here if f star is the optimum function, then it is optimum with respect to all that mean f_1 , f_2 and f_m then this solution is called the ideal solutions. And all the objective vectors, these all the objective vectors are called ideal objective vectors.

(Refer Slide Time: 04:47)



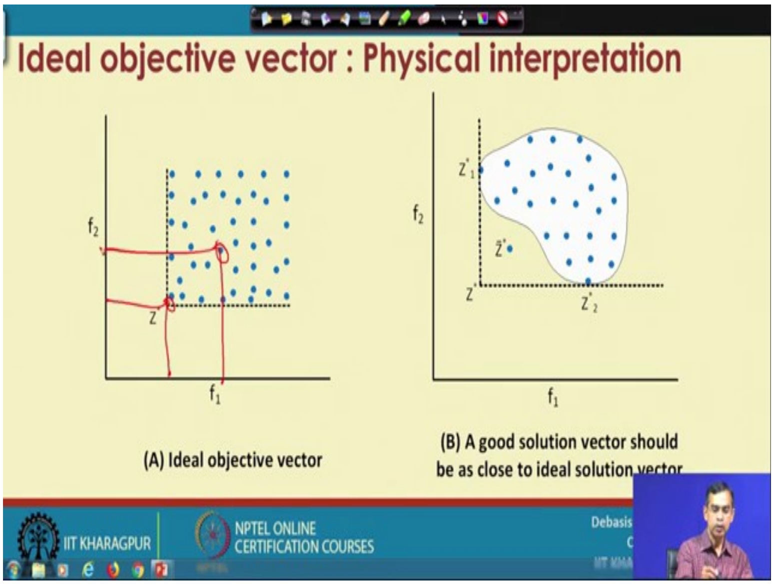
Now, the same thing can be discussed with a visual description and illustration right. Now, here suppose f_1 and f_2 are the two objective function. And we assume that f_1 is to be minimized and f_2 is also to be minimized for simplicity. Now, so, in this is the search space.

(Refer Slide Time: 05:13)



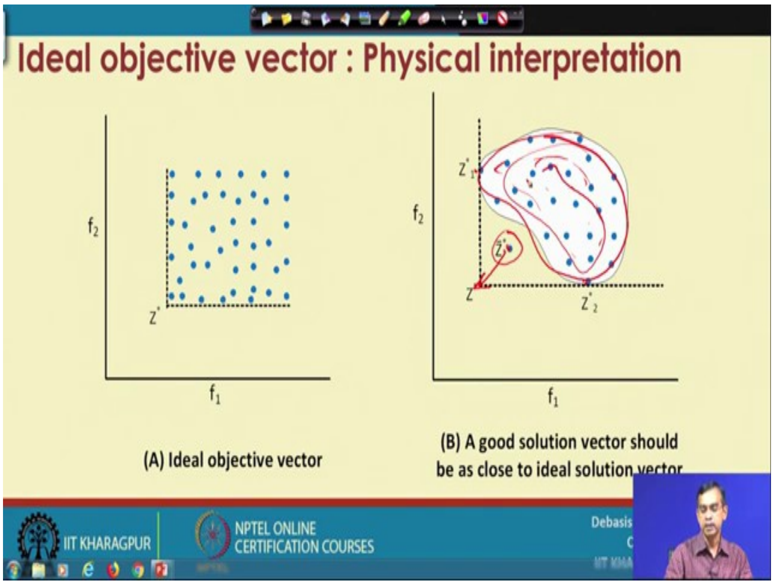
This is the search space for the entire what is called the searching of towards the optimum values. Now, out of this search space, we can see there is one solution.

(Refer Slide Time: 05:28)



This one which is basically satisfying all the objective functions simultaneously. So, if this is the solution then it is f_1 is minimum and f_2 is minimum. Now, if you consider any other solution we can say that this is not the solution which simultaneously satisfying this one. For example, if it is minimum f_1, f_1 is also not minimum, then f_2 is also not minimum, so like this. So, the solution point which basically signifies this one, it is basically corresponding to the ideal objective solution. And if there is an objective function which exist there then it is called the ideal objective vector. So, this is the concept it is there. Now, so this solution as we have discussed about it is called the ideal objective solution.

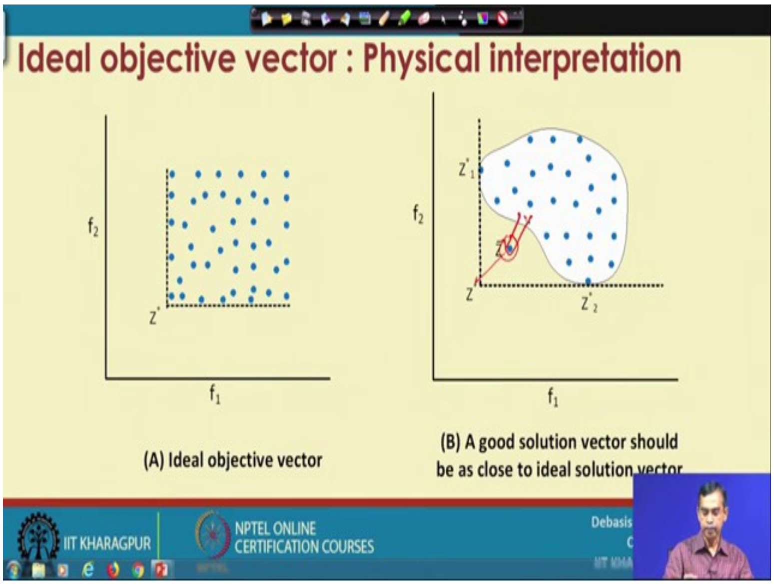
(Refer Slide Time: 06:27)



Now, in this in this figure, if we see it again, we can see that this is a one solution; this is the one solution. This solution may be minimum with respect to f_2 but the solution is not minimum with respect to a f_1. Similarly, this solution is minimum with respect to f_1, but not with respect to f_2. However, if we can find one solution like this which is obviously not exist here then we can say this is the one solution which is minimum with respect to both f_1 and f_2.

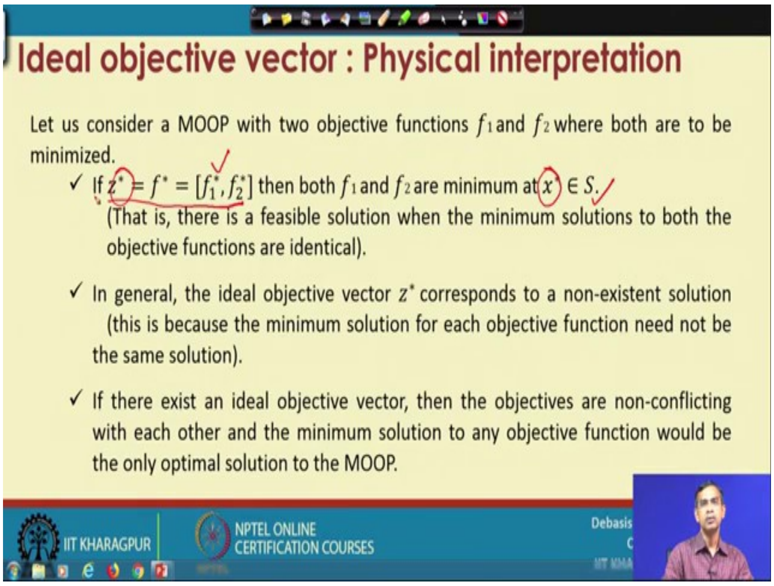
Now, so, this is the case if we have, then say suppose this is the one search space that means, we find many solutions in this region. Then definitely this is the one solution this is the trade off solution; that means, if it is good with respect to f_2, but not good with respect to f_1 and vice versa. Now, if there is any other solution which is very close to this ideal solution, then we can say this solution is more preferable than any other solution in the search space.

(Refer Slide Time: 07:41)



So, while we are searching if we find one solution which is not necessarily and ideal solution, but very close to the ideal solution right then this solution can be considered as a solution of our objectives. So, this is a desirable solution. So, a good solution, so vector should be as close as to ideal solution vector. So, this is the interpretation that we can have from this concept ideal solution ideal objective solution.

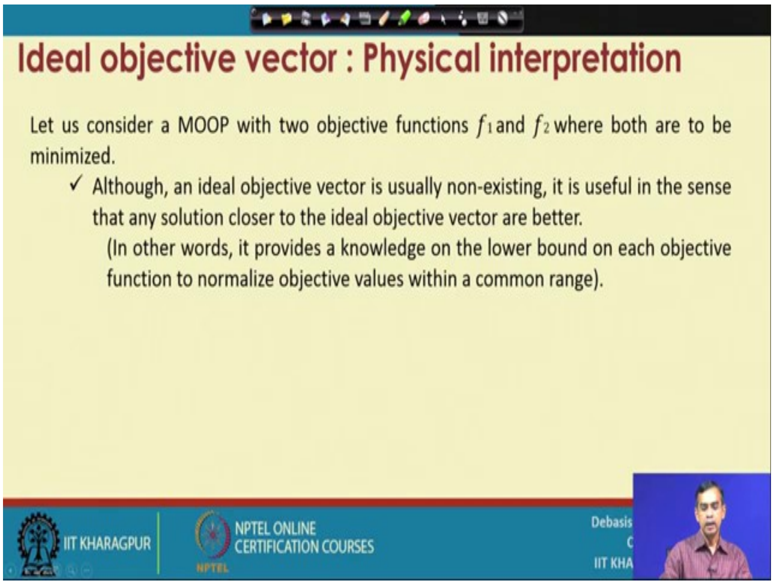
(Refer Slide Time: 08:10)



So, here now we can generalise little bit precise about say suppose there is a multi objective optimization functions with two objectives f_1 and f_2 where suppose both are to be minimized, both are to be minimized. Now, so if there is a solution we say that the solution is $f_1^{¿}$, $f_2^{¿}$ then both f_1 and f_2 are minimum at $x^{¿}$ that is the solution space belongs to the search space. So,.So, in general the ideal objective vector. So, this is basically the ideal objective vector.

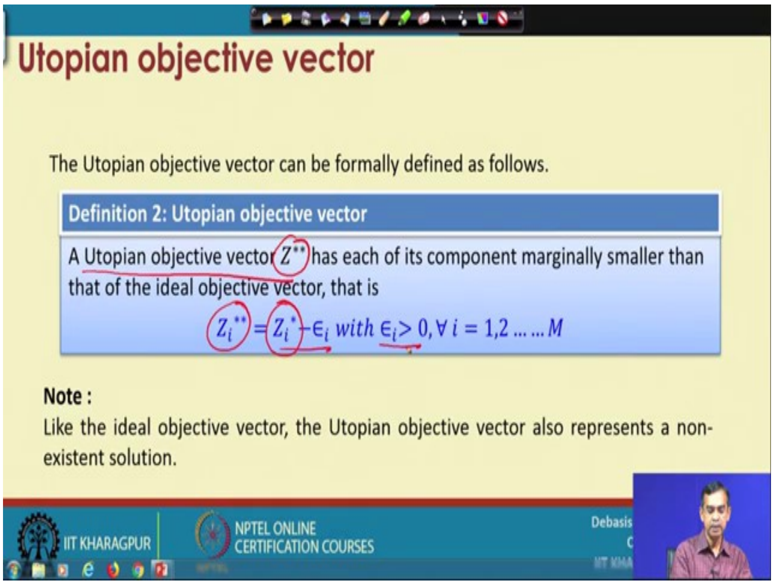
So, $z^{¿}$ we can say it is an objective vector which is ideal right, right in fact, the ideal objective vector corresponding to a non-existence solution that is why we called it is an ideal. Because many multi objective optimization problem are conflicting objective it is it is very rear it is in fact, impossible to see one objective function which is one objective vector which is minimum with respect to both the objectives. And if there exist an ideal solution then the objectives are non-conflicting with each other, and then minimum a solution to any objective function would be any optimum solution to the problem. So,. So,. So, this is the concept of ideal. So, ideal objective vector is non-existing one solution we can say.

(Refer Slide Time: 09:51)



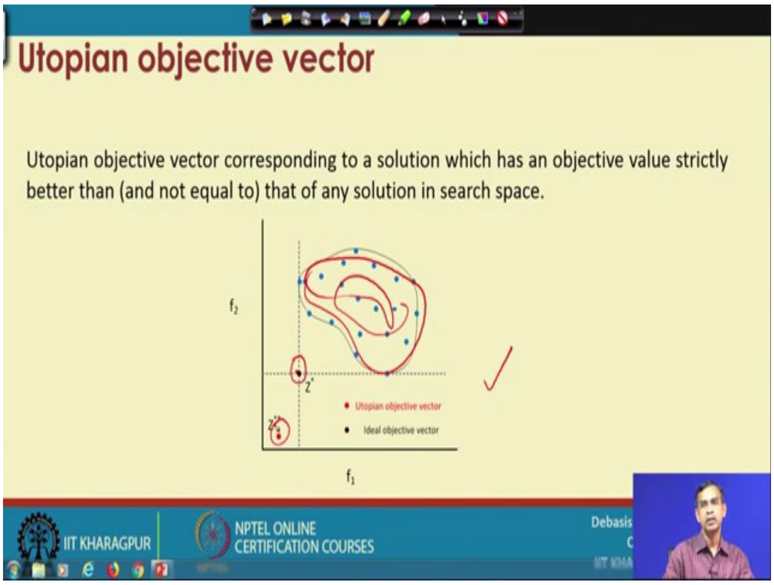
Now, so, fine now so, we have learned that ideal objective solution ideal solution is usually non-existing, but it is useful in the sense that any solution closer to the ideal objective vectors are preferable. So, this is the usefulness of this concept of ideal objective vector.

(Refer Slide Time: 10:18)



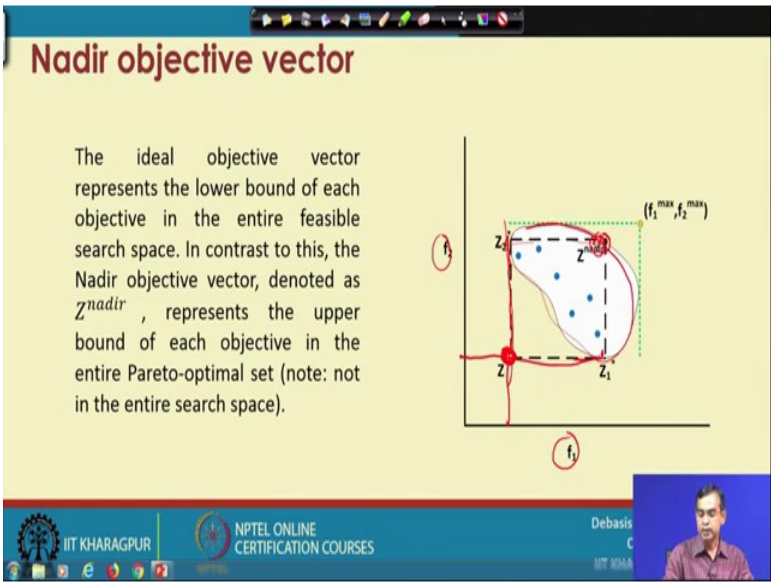
Now, we will discuss about another solution such a solution is called the utopian solution. Utopian means it is it is it is a is a fictitious, it is never possible like an ideal also, but it is an another extent. Now, let us define how the utopian solution can be. A utopian objective vector, so we denote it at Z^{**} has each of its component marginally smaller than that of the ideal objective vector. So, if Z_1^{*} is the ideal objective vector then Z^{*} will be far better than ideal objective vector. So, it is basically with epsilon i greater than 0, so that means, Z^{*} Z_1^{**} is far better than Z_1^{*} then such a solution is called the utopian solution. And then solution vector if it is possible then it is also called the utopian objective vector.

(Refer Slide Time: 11:26)



Now let us illustrate the concept with a with a with an illustration. Now, here this figure is planned to illustrate the concept of utopian objective vector. Now, now this is the search space this is the search space. And we have learned that this is the one solution which is called the ideal solution. And then utopian solution is far better than the ideal solution mean this is the one solution is a utopian solution. And the vector for which satisfy the solution is called the utopian objective vector. Now, so we have learned about the ideal solutions and thereby the utopian objective vector.

(Refer Slide Time: 12:08)



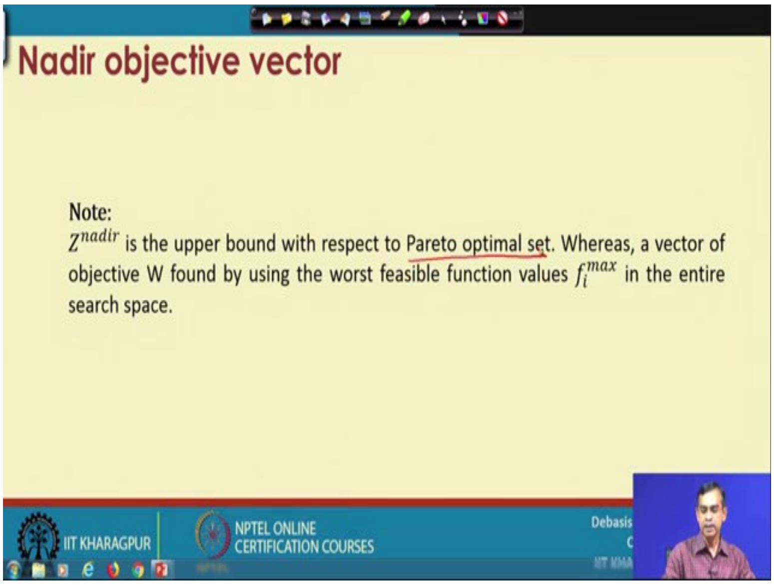
Now, as we have learned about it like say like ideal solution is non-existing solution for conflicting objectives, similarly the utopian solution is also a non-existing solution when the objectives are conflicting of type. So, this the two solutions types, one ideal solution and utopian solutions. Now, we learn about another solution it is called the Nadir solution.

Now, let us see what exactly the Nadir solution is ok, we can explain this concept by means of an example of an example. Say this is the solution search space all solutions possible. And both f_1 and f_2 are to be minimized for the sake of generality. And then so this is basically in our case it is the ideal solution. And then there is one $Z^{¿}$ solution which is basically minimum with respect to f_1, similarly there is another solution $Z_1^{¿}$ which is minimum with respect to f_2.

Now, so for this $Z_2^{¿}$, we can see the f_1 is maximum ; and for the $Z_1^{¿}$ we can see the for the $Z_1^{¿}$, f_1 is maximum value; and for the $Z_2^{¿}$ also f_1 is maximum. So, this is the one solution if we say which is on the boundary or basically close to the solution then there is an extreme solution such an extreme solution is called the Nadir solution. Now, so, so nadir solution is like this. So, it is basically the extreme values with respect to the optimization objectives.

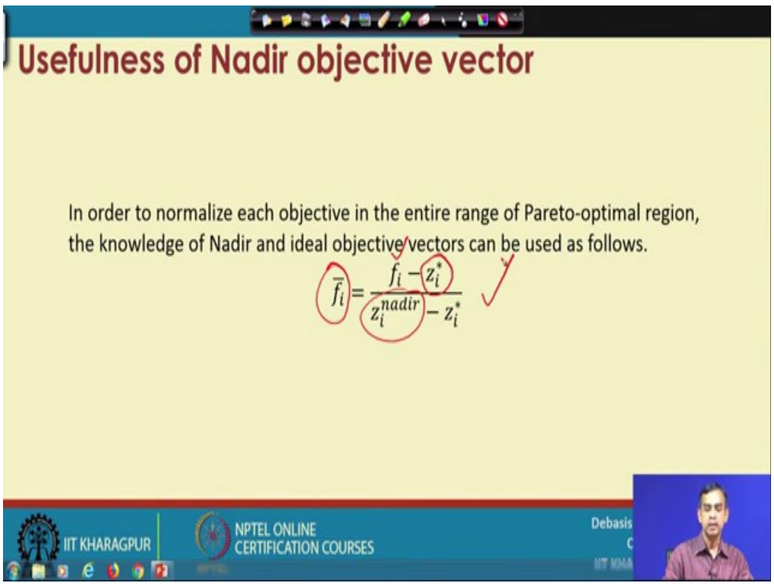
And then and there is another solution suppose this one and this is the another solution is this one these are the basically again another extreme solutions in the solution space. Now, extreme solution with respect to we say it is f_2 and this is an extreme solution with respect to f_1. And then there is another solution which is like this having this satisfaction then this solution is just like a utopian solution in case of ideal objective, it is basically the another utopian solution with respect to Z^{nadir}. Now, so this basically gives an idea about how the scope or range of the solutions maybe there. Now, so, this is the idea about the nadir objective vector.

(Refer Slide Time: 15:07)



And, so, in a simple word, the nadir objective solutions is the upper bound with respect to the set of all optimum solutions. We will turn all the solution as the Pareto optimum solution. We will term them as a Pareto optimum solution. So, it is basically the upper bound in that sense now. So, having this is a concept we will just see exactly what is the usefulness of these solutions ok. We have learned about the usefulness of ideal objective vector. Now, similarly the there is an application of the nadir objective solution for a given this one.

(Refer Slide Time: 15:49)

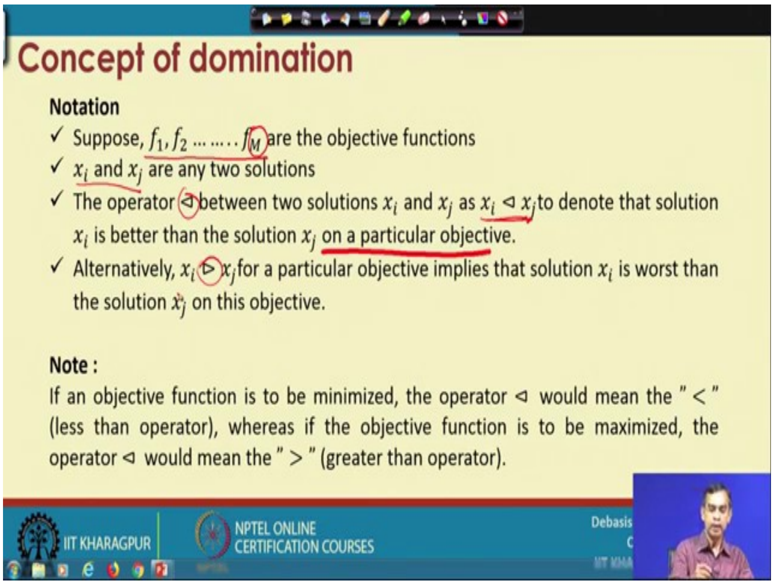


So that that can be explained by this form. So, this solution the nadir objective solution usually used to scale the objective vectors the scaling of objective vectors we have learnt while we are discussing about scaling in case of single objective genetic algorithm. Now, so this is the one idea here. If $z_i^{¿}$ is the is the ideal objective solution and f_i is the any objective vector and the z_i^{nadir} is corresponding to the nadir objective solution then the scaled objective function can be denoted by this. And it is satisfied this formula. So, this is the one what is called the idea where the ideal objective solution and the nadir solution is used. So, it is basically used to scaling the objective vectors.

(Refer Slide Time: 16:53)

So, we have learned about the different solutions that may have there in with respect to two objective functions we have discussed, but that idea can be extended with respect to multiple objective factors as well as. Now, we will discuss about another important concept, it is called the concept of domination.

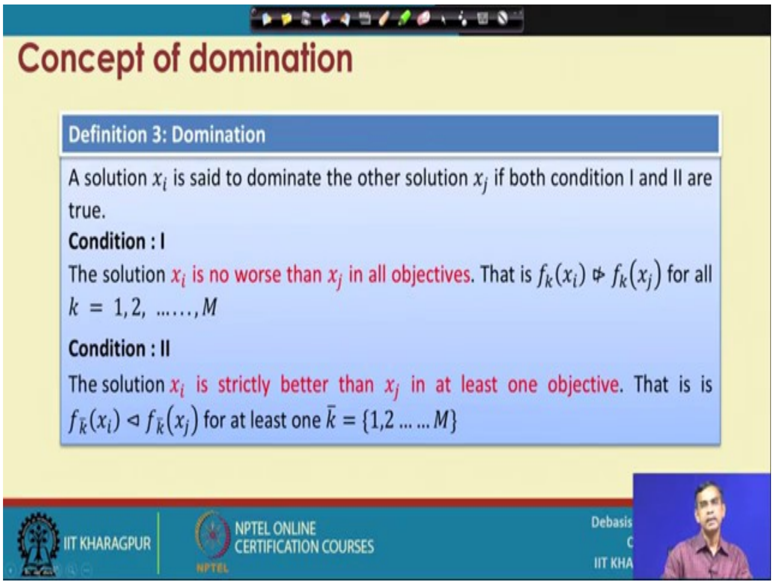
(Refer Slide Time: 17:16)



Now, concept of domination as I told you so there are some objective functions which are conflicting ok. Then let us suppose the M so there are M number of objective functions and there exist any two solutions say x_i and x_j are any two solutions in the search space. Now, we can define a symbol it is basically an operator between two solution x_i and x_j to denotes that $x_i dominates$ x_j to denote that solution x_i is better than the solution x_j for a particular objective on a particular objective with respect to one particular objective. Similarly, if $x_i dominates$ x_j or $x_j dominates$ x_i this kind of symbol can be used.

Now, here our objective or the discussion that we are going to discuss about that if two solutions are given to you, then how you can decide that the one solution dominates other solution or no one dominate each other. So, dominate in the sense that one solution is better than the other solution. So, this concept is called the concept of domination.

(Refer Slide Time: 18:43)

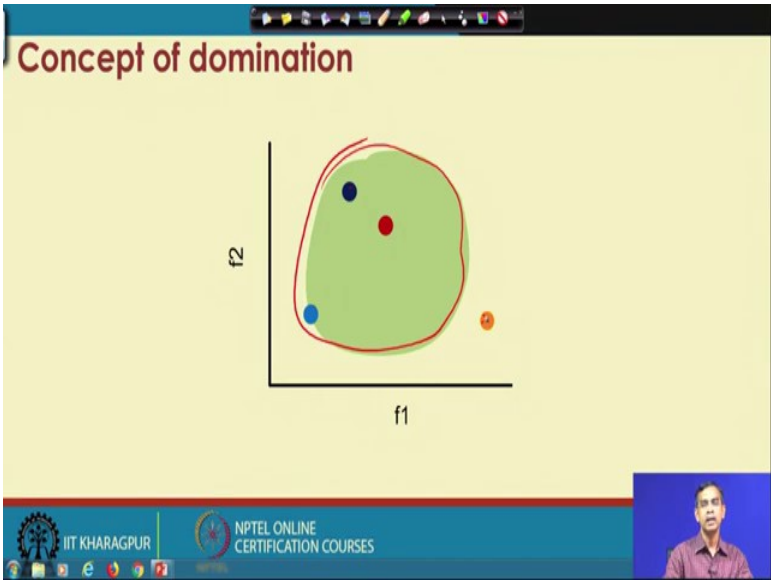


And it basically so there is a precise definition for the domination. Supposed two solutions x_i and x_j are there, then we can say that that solution x_i dominates another solution x_j if they satisfy two conditions. So, both the conditions are to be satisfied, then only we can say that x_i dominates the solution x_j. In other words in a simple words x_i is a better solution then x_j.

Now, so, the condition first. The first condition is that x_i is no worse than x_j in all objective that mean there are objective function for all objective function k, where $k=1$¿M and then that solution does not dominates any other solution. So, it is basically the first condition; that means, we will illustrate an example anyway. So, this is very important one statement that x_i is no worse no worse then x_j in all objectives in all objectives. So, these are the things to be noted.

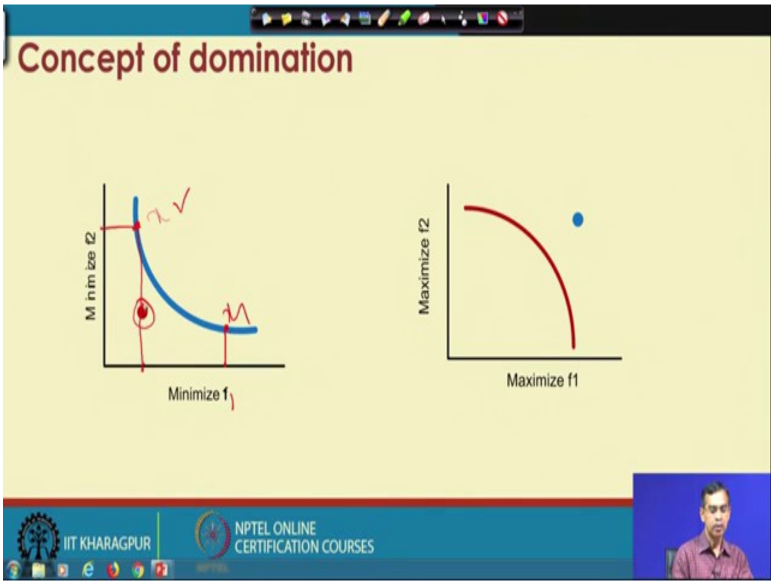
And the second condition is that x_i is strictly better than that means, it is better than always in at least one objective. So, it is basically again this point to be noted is strictly better than and at least one objectives. So, if these two conditions are satisfied between x_i and x_j then we can say that x_i is x_i dominates the solution x_j.

(Refer Slide Time: 20:45)



Now, let us say let us see the example so that you can understand about it. Now, in this example $f1$ and $f2$ are the two objectives. And so these are the solution space. This is also another solution which is not in the search space or whatever it is there. Now, out of this solution this and these we have to check that which solution dominates which other solutions. Then definitely our objective is to select those solutions which dominates all other solution if there exist one solution which dominates all other solution that will be our desirable solution. Now, let us see what are the different situation may occur, so that we can understand which solution dominates other or how the dominant property is can be this one can be satisfied.

(Refer Slide Time: 21:36)

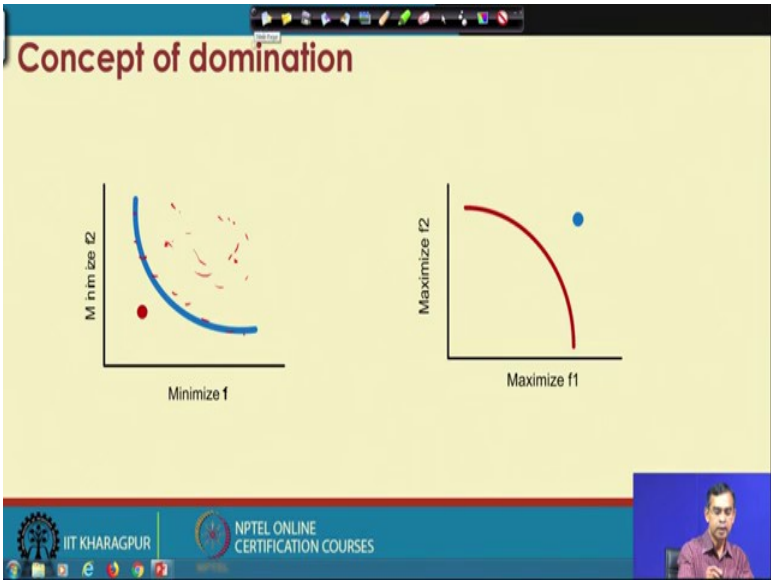


Now, this is the diagram that I have planted it, so that you can understand it. And I am discussing about with respect to the different what is called the different objectives type. For example, here this is minimize $f1$, this is minimize $f2$, this maximize $f2$, and like this.

Now, anyway so if we see in this slides we can understand that this is the one solution, and this is the one solution right. Now, this solution and this solution if we see so this solution so far the $f1$ is concerned right $f1$ is concerned it is worst or it is not good as the this solution $f2$ is concerned. Because this solution $f2$ has the better with respect to $f1$. On the other hand, this solution is also good with respect to $f2$, but not with respect to $f1$. However, this is the one solution, which is good with respect to both the solution as well as this solution.

So, we can say that this is the one solution which dominates this solution. On the other hand, this solution or this solution, so if this is x_1 and this is x_2 then we can say that neither $x_1 dominates\, x_2$ or $x_2 dominates\, x_1$. Because the condition one and condition two are not satisfied for the solution x_1 and x_2 . However, condition one and condition two both the conditions are satisfied for this solution. Now, so, in this context what you can say is that this solution dominates these are the solution whereas x_1 and x_2 does not dominate each other.

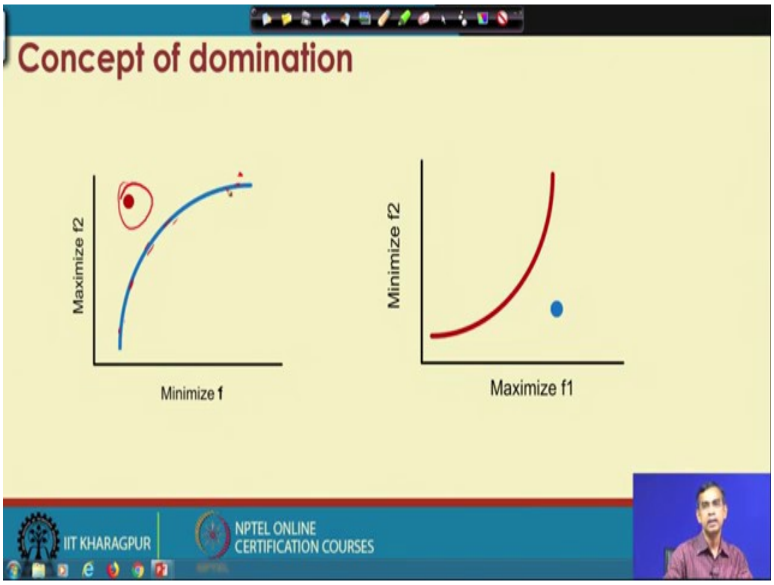
(Refer Slide Time: 23:34)



Now, similarly all the solution which lies on this region they are basically non dominating, non dominating no one dominates other. But any other solution or any other solution here also we can say this solution is better than this solution in some respect. So, we can say that all solutions dominate all other solutions here. So, this concept can be can be understood from if we verify the two properties.

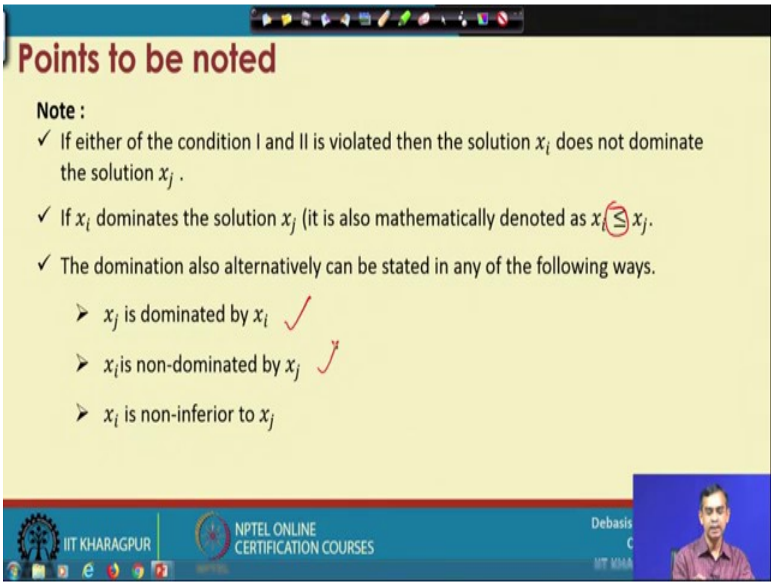
Similarly, in this example the same as this one, but is a maximize $f1$ and $f2$ we can extend the same idea for this two solution. And then we can check that these are the solutions which do not dominates one solution. And this is the one solution which basically it is dominates any more solution in this one. So, this is basically a dominating solution and these are the non dominating solution fine.

(Refer Slide Time: 24:38)



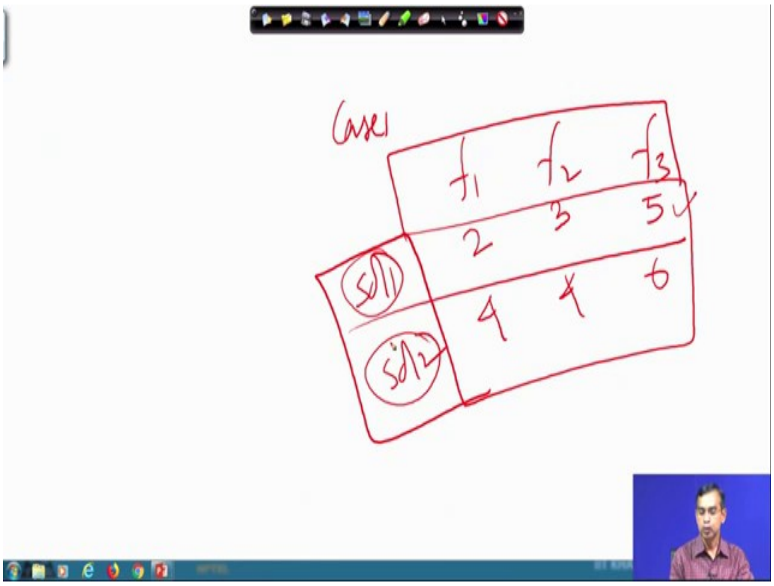
So, the same thing can be extended for the other minimize and this maximize, and this is maximize and this minimize. And these are the solution is basically these are the solution we can these are the solution is basically dominates any other solutions here, but all the solutions is lies on this region are non dominating, the same thing it is there. Now, so, this is the concept of domination if you want to understand more precisely about this concept, then I can plan one example.

(Refer Slide Time: 25:13)



Ok. So, let us illustrate the concept of domination because it is very important and then it is we have to we will explain the three cases, so that you can understand the concept it is there. So, case one, and suppose f_1 , $f_2,\wedge f_3$ are the three solutions are the three objective vectors. And there are two solutions one and solution two.

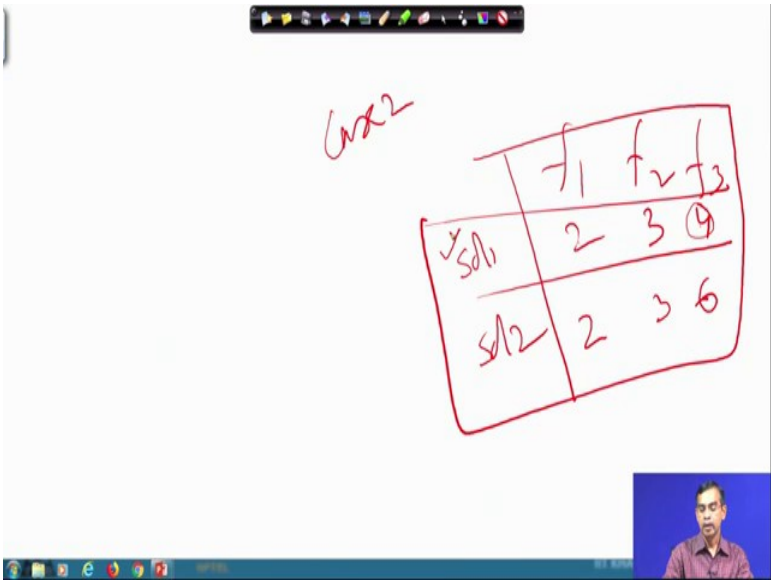
(Refer Slide Time: 25:47)



Now, say solution one is suppose both are to all these objective vectors are to be minimized. And say solution one has the results say 2, 3 and 5; and then solution two has the result 4, 4, 6. Now, in this case ok, so f_1 , $f_2,\wedge f_3$ are all to be minimized. Now, if we consider so then this solution one and solution two, if we compare then solution one condition. So, condition one so far this is a it is no worse than any other objective function.

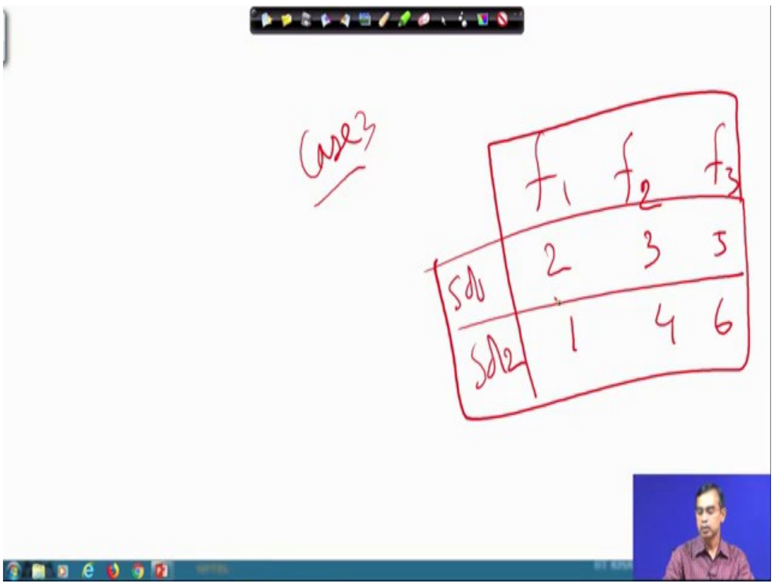
So, the condition one is satisfied for the solution one and then also with respect to at least one this solution has the satisfied right so that means, in this case both 4, 4, 6, and here 2, 3, 5 with respect to all these solutions satisfy this one. Then we can say that this solution one dominates the solution two.

(Refer Slide Time: 27:08)



Now, another example so it this case two same thing earlier so, again $f_1 \wedge f_2 \wedge f_3$ are the three objectives or which are to be minimized. And there are two solutions solution one. And suppose solutions one is 2, 3, 4; and solution two is 2, 3 and 6. Now, in this case, so solution one is no worse than any solution there in solution two. And it solution two and three, and solution for f_1 with respect to f_1 and f_2 both the solutions are same, but at least with respect to f_3 this solution is better than this solution. Then we can say again here solution one and solution two solution one dominates solution two.

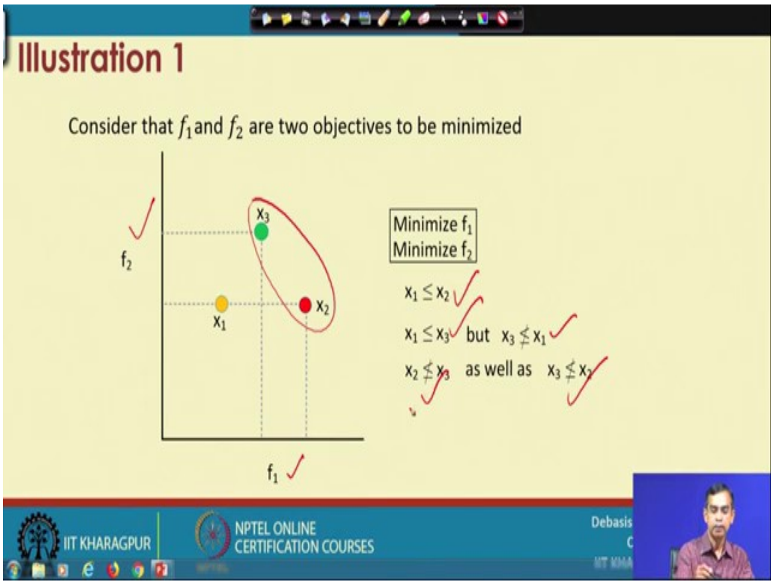
(Refer Slide Time: 28:00)



Now, let us see another case three. Again we will consider $f_1, \wedge f_2 \wedge f_3$ both are to be minimized. And there are two solutions one and solution two. And let the solution one, 2, 3, 5, and 1, 4, 6. This kind of, this kind of solutions are there. Now, here we can see ah. So, solution two is better with respect to a f_1 . However, it is no it is worse with respect to both $f_2 \wedge f_3$. So, this means that neither solution one and solution two dominates that means, solution one neither dominates solution two or solution two does not dominate solution one. So, this is the example that can be helpful to understand the concept of dominance there.

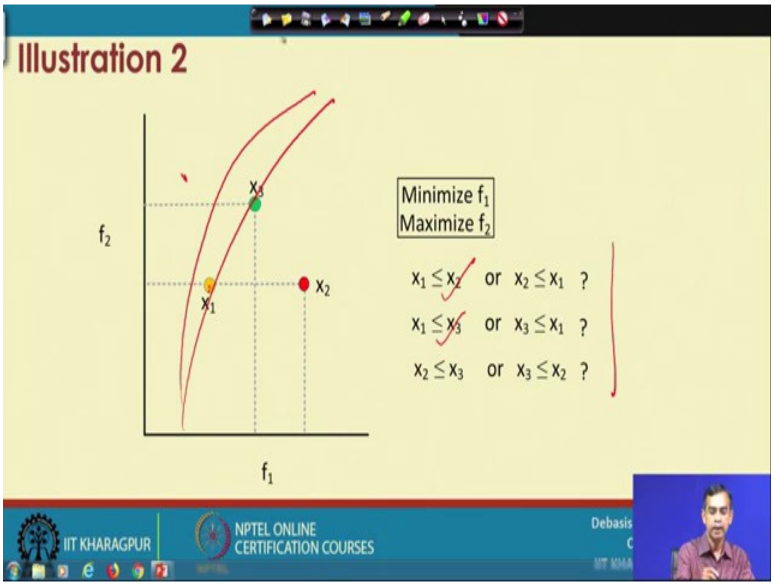
Now, so we have learned about the dominant the concept of domination. Sometimes many languages that can be used to represent or notation also. So, here if x_i dominates the solution x_j , then we can write this kind of symbol also used for brevity. And also it can be termed as x_i is dominated x_j is dominated by x_1 or x_i is non-dominated by x_j , or x_i is non-inferior to x_j this kind of solutions are there any way. So, meaning whatever the way we can express it the concept is like this one.

(Refer Slide Time: 29:40)



Now, we can illustrate the same concept two more slides are there as you have discussed already, you can verify yourself that. Here x_1 and x_2 , x_1 dominates x_2 . Bow, similarly x_1 and x_2 if we consider then x_1 dominates x_3 but if we consider x_2 and x_3 then neither dominates this one. So, here we consider f_1 are to be minimized and f to are to be minimised that you can verify yourself. So, this is the thing that we have written here and these are the things you can verify yourself.

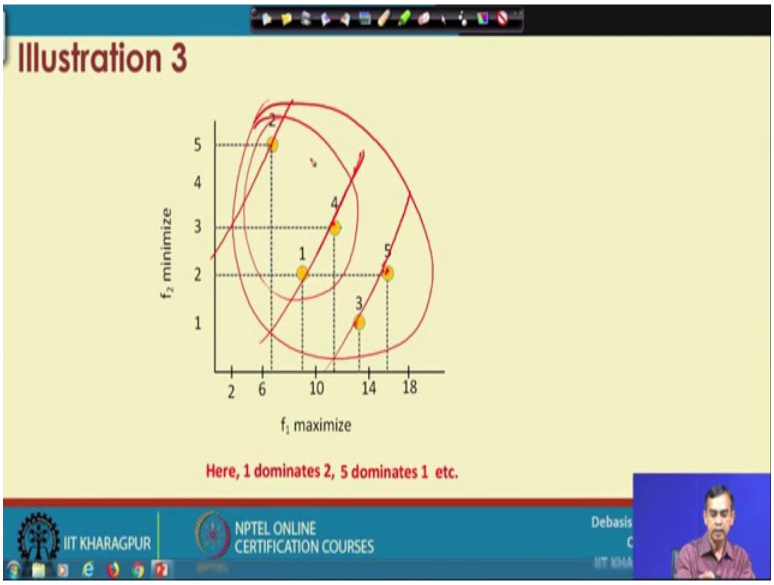
(Refer Slide Time: 30:15)



Now, second illustration here just ok. this is minimize f_1 and maximize f_2 . So, here also the ok, so it is basically minimize and maximize that means, the objective that solution the solution it is basically it is like this or whatever it is there. So, if we consider x_1 and x_2 because it is a minimize and maximize the solution region will be look like this one. So, all the solution which is on this region they are basically non-dominating each other.

However, any solution if you consider this one it is preferable and optimise this one, so this way we can verify that x_1 and x_2 , x_1 dominates x_2 in this case, and then x_1 also dominate x_3 right and. So, so, but x_1 and if you consider x_2 and x_3 , they are not dominating each other so that you can verify and then understand about it.

(Refer Slide Time: 31:21)



And similarly this is an another case where the f_1 is maximize, and f_2 is minimize. So, the idea it is that it is like this. So, you can say that is there. So, any solution not dominates this one, but here if we consider this solution all right or these are the solutions which dominates all other solutions. So, it is like this. So, all solutions dominate any solution which is on this range on this line or so similarly.

So, we can say that if this is the search space then all the solutions are dominated by all the solutions, but here no solution dominated by this one and then the concept is like this anyway. So, it requires little bit thorough checking and then you can understand that which solutions dominates which of the solutions ok.

So, this is the concept of domination. And then we will discuss about important properties this domination solution holds good that will be covered in the next lecture.

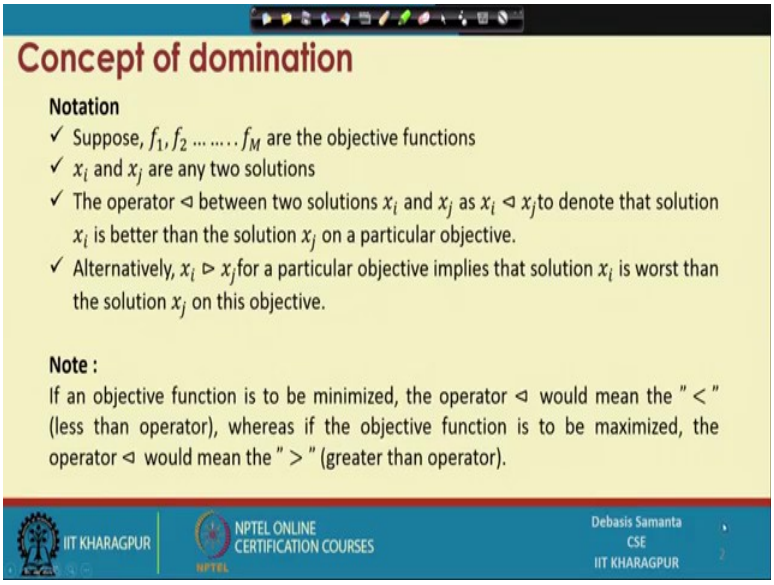
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 26
Concept of Domination

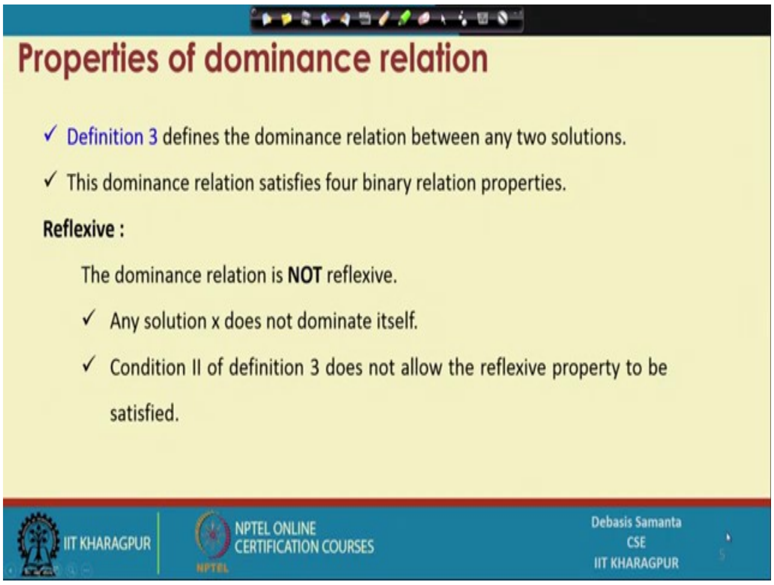
In the last lecture, we are learning about two things the concept of the solutions. So, far the multi objective vectors are concerned, and then we discussed about concept of domination. In this lecture we will try to learn about the properties that the concept of domination holds and then we will discuss about the pareto optimal font which is a very important concept so far the multi objective solution is concerned.

(Refer Slide Time: 01:03)



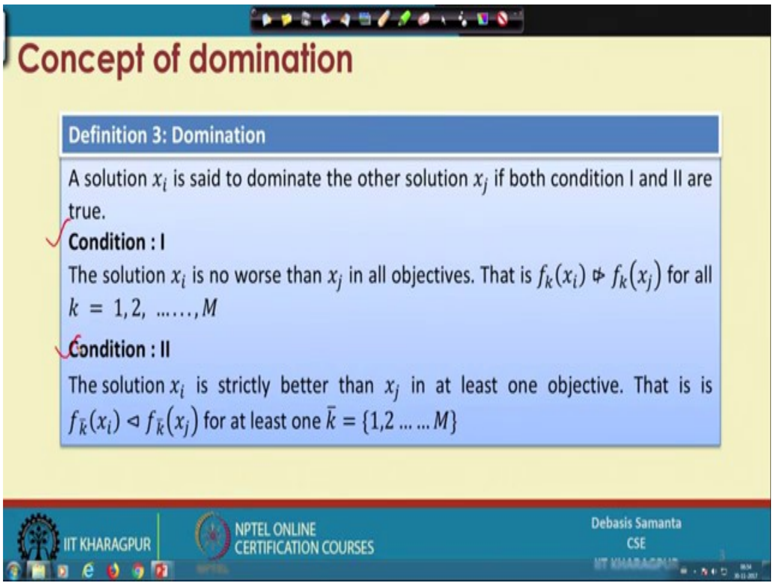
So, now, this is in continuation with the previous discussion we just introduced the concept of domination whether two solutions x_i and x_j with respect to M number of objective vectors are dominating or not. Now, today we will discuss about the properties that this dominations relation holds. That means, if there is a two solutions x_i and x_j where x_i dominates x_j then what kind of relations that it can holds.

(Refer Slide Time: 01:39)



Now, we can start with the concept it is here. So, definition 3, the definition 3 we have the definition see that we have discussed in the last lecture we say that two solutions are said to be the or two solutions x_i and x_j , if it is there then x_i said to dominate x_j . If two conditions condition 1 and condition 2 which is stated here, satisfied there.

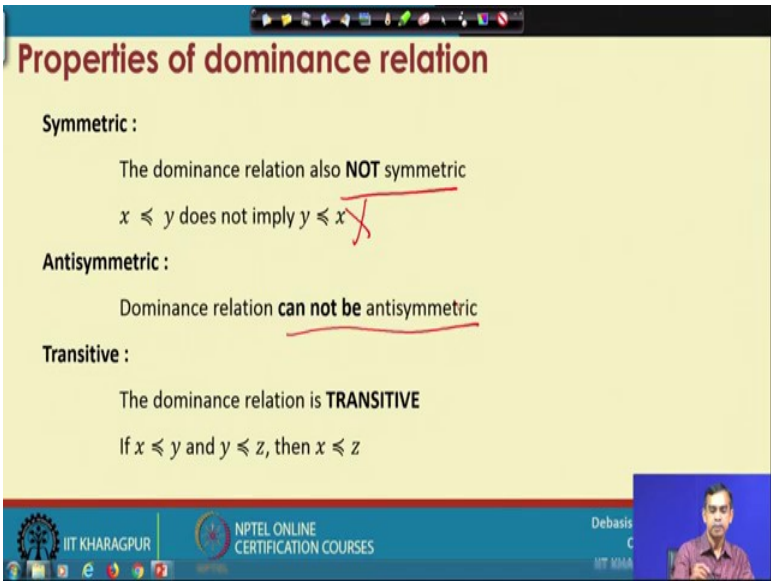
(Refer Slide Time: 01:44)



Now, let us see if there exist two solutions and then what relations that they hold good. Now, so, it required basically it is basically the relation between the two solutions. So, it is the binary relations. We can say and we know the binary relation concept it is there and that relation can reflexive can be symmetric or can be transitive, and we can say that the solutions x_i and x_j is the dominance relation is not reflexive that mean a any solution x does not dominate itself. So, this is a one important concept that it is not reflexive. Condition because, so condition ok.

So, so far the condition one is concerned x_1 , x_i and x_i , the first condition holds good, but the second condition that it should be strictly better than with respect to at least one which is not holds good there. So, that is why it does not satisfy the reflexive property. So, that is why this relation is not a reflexive relation.

(Refer Slide Time: 03:24)

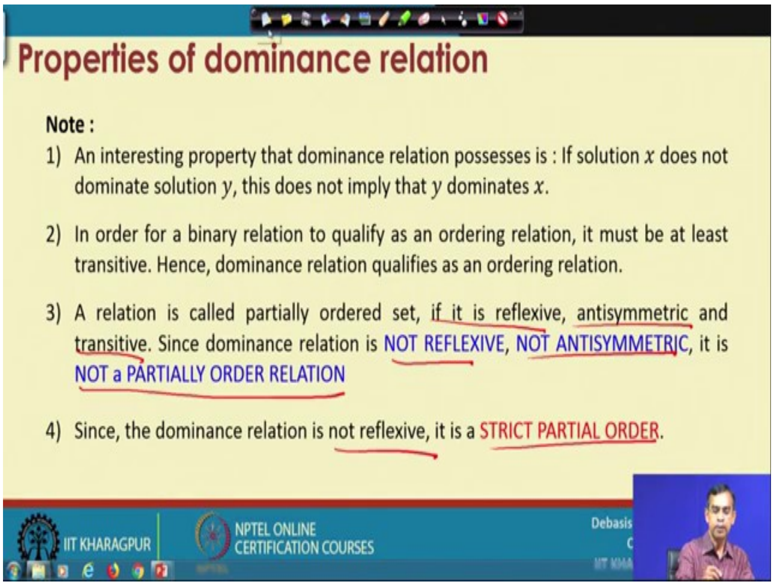


Similarly, this solution is also not symmetric relation, not symmetric relation, so not symmetric relation. So, means that x , if x , if $x \, dominates \, y$ it does not imply that $y \, dominates \, x$. So, it is not a symmetric relation.

Now, it also can be anti-symmetric; that means, dominance relations if x does not satisfy y and then y can be also can be also dominate this x also that is why it is called the cannot be anti-symmetric. However, it satisfy one property it is called the transitive property that mean if $x \wedge y$ two solution. So, that $x \, dominates \, y$ and similarly there is a solution z such that $y \, dominates \, z$ then we can say that $x \, dominates \, z$. So, it is called the transitive property.

So, what we have understood is that, so dominance relation is not reflexive, not symmetric, not anti symmetric, however it is transitive.

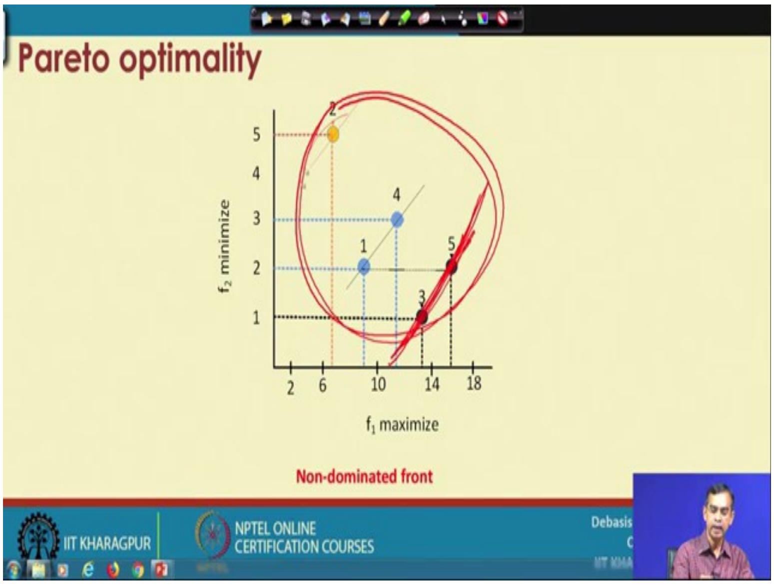
(Refer Slide Time: 04:49)



Now, so, based on this concept a relation a binary relation are a binary can be termed as a partially ordered set, partially ordered set if it is reflexive, it is anti symmetric and transitive. So, in this case as it is not reflexive, not anti symmetric then it is not a partially ordered set. So, that is why it is called is not a partially ordered relations. However, seen it is not reflexive and then it basically strict strictly partial order. So, the concept the domination relation is therefore, not a partially ordered relation it is a strict partial order relation.

So, this is a property that the domination relation satisfied.

(Refer Slide Time: 05:49)



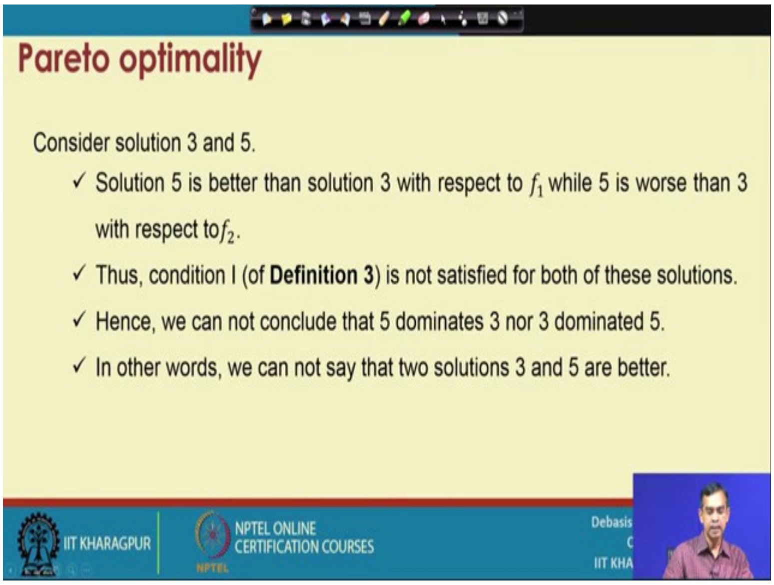
Now, we will discuss about pareto optimal solutions, the concept of pareto optimal solution. Now, before going to discuss these things we will continue our discussion of the concept of domination where we have discussed about the two objective PPTs here then how they can dominate each other or this kind of things here. Now, in this figure we can say this 3 and 5 when a f_1 is maximise and f_2 is minimise we can say that 3 and 5 neither dominate each other; that means, this solution 3 does not dominate solution 5 or solution 5 does not solution 3.

However, if we consider solution one solution then we can say that this solution one or the solution 3 dominates 1 or solution 5 dominates 1 or solution 5 dominates both 1 and 4 solution 3, this one. Now, here all the solutions which lies on this line is basically not dominated by any other solution; however, they dominate all other solutions these and this one all the solutions. So, these are the solutions dominates all other solutions in this region, and as there is no solution here we can say that this solution is not dominated by any other solution. Now, the solution set which lies on this line like is called the optimal solution set or non-dominated solution and these are the front it is called the non-dominated solution front.

So, this is the concept it is there. Now, solution, so these are the optimal solution with respect to our searching of multi objective solution and ok, so this is the entire search space then all the solutions are desirable solution because they are at least better with respect to at least one objective vector if it is not there.

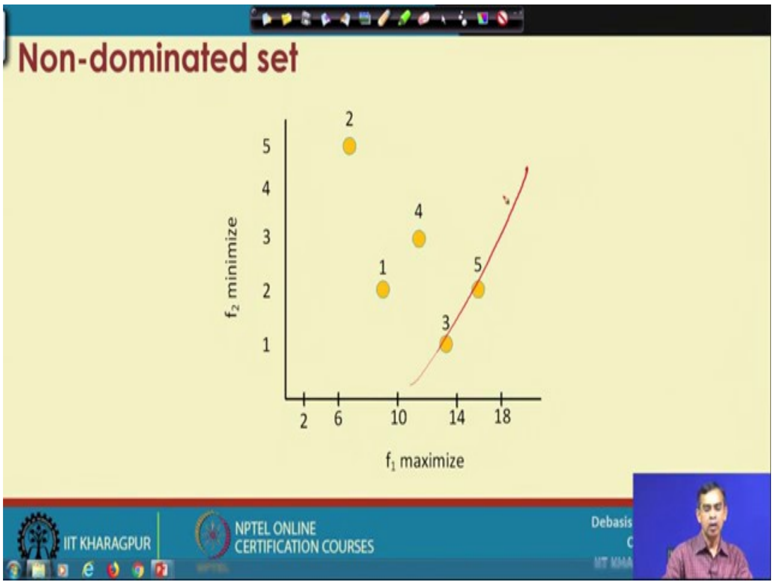
Now, so this is a concept it is there and we will see exactly the pareto optimal solution if all the solutions that is this is the entire search space that is possible or all feasible solution those are there and out of these all feasible solutions if this is a solution that can be on the non-dominated front then we can say that this is the pareto optimal front. This front is also called pareto optimal front. So, the condition is that the solution will be termed as pareto optimal only the entire surface is covered there. So, this is the concept of pareto optimality and then we will discuss about the pareto optimal front.

(Refer Slide Time: 08:28)



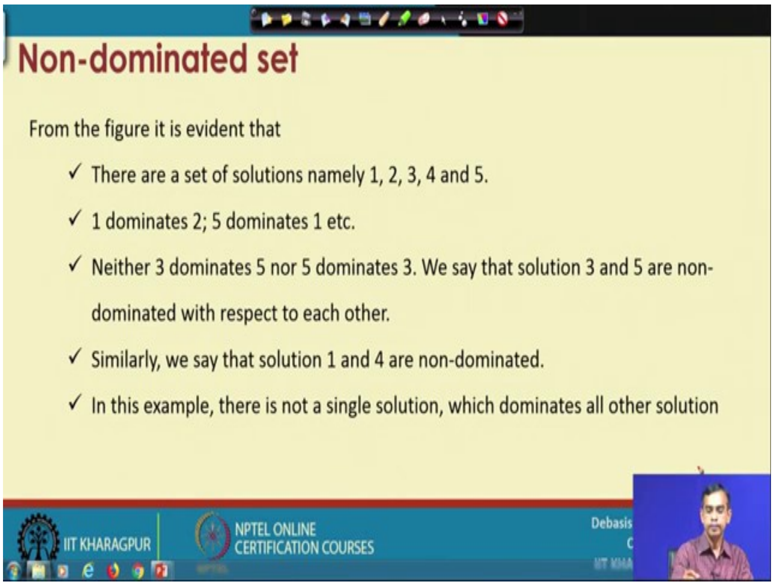
So, so that idea that we have discussed it is basically the same concept.

(Refer Slide Time: 08:36)



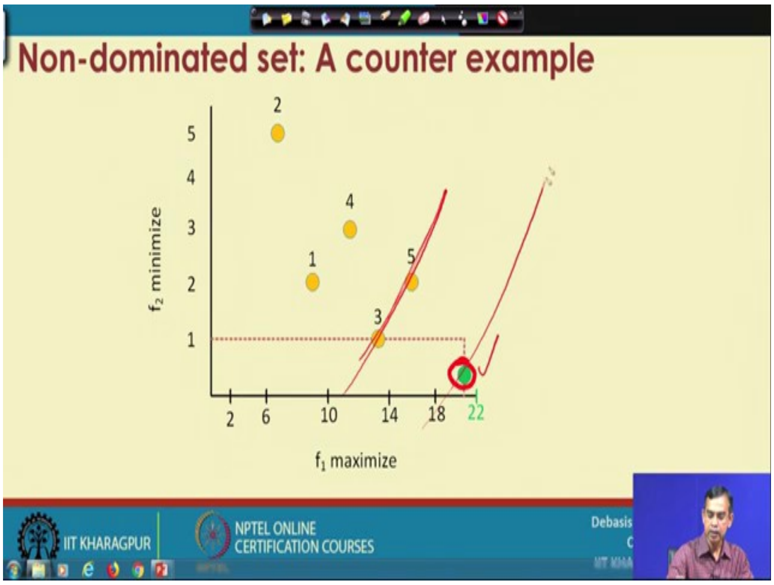
And so, again, so this is the front that front can be termed as a non-dominated set front that we have already discussed.

(Refer Slide Time: 08:50)



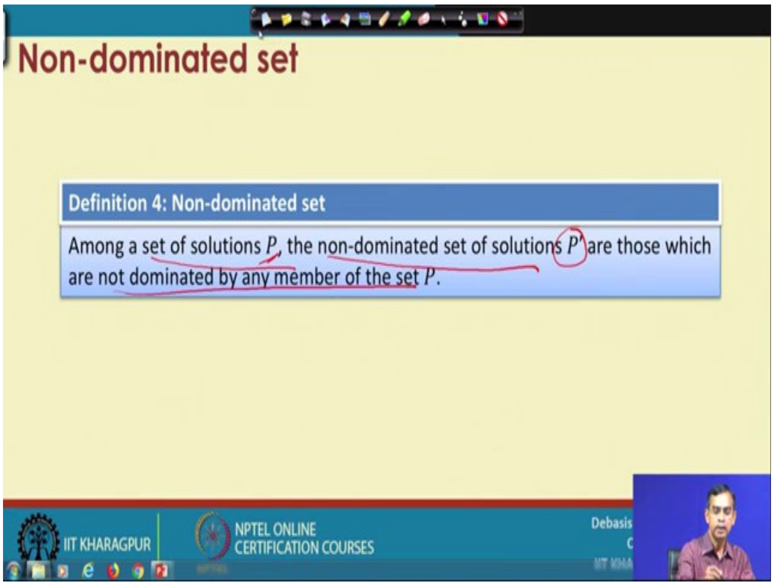
And. So, we if we check it all these you can verify with respect to the previous slides we can verify there are a number of solutions the solutions are 1, 2, 3, 4, 5. 1 dominates 2, 5 dominates 1 like this kind of concept is there and then we can see that there are some solutions 3 and 5 which are non-dominated solution ok. So, that you can check it and you can find that these are the conditions all conditions hold good.

(Refer Slide Time: 09:27)



Now, so, non-dominated as I told you, if this is the solution is a non-dominated and if we can find any other solution then they are no more non-dominated solution. So, this solution is become the dominated. So, this is basically another dominated front actually ok. So, this is an example when they are not the dominated like this if there exist some solution is here.

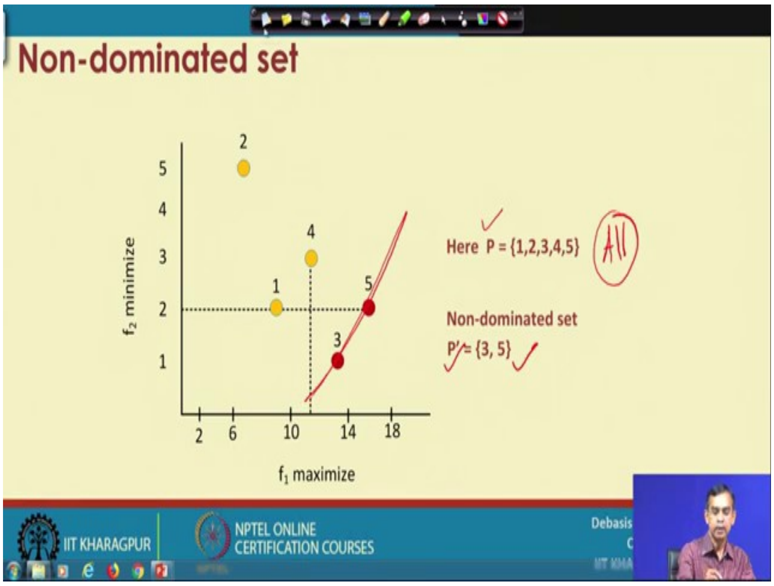
(Refer Slide Time: 09:52)



So, now, we can precisely mention about when a solution will be termed as a non-dominated solution or all solutions because they are not a single solution which can be there, there may be multiple solutions also, all solutions which are non-dominated they are called the non-dominated set.

So, it is like this idea, a set of solution all solutions P then non-dominated set of solution there is a subset P' of P are those which are not dominated by any member of the set P. So, it is a concept of non-dominated set. And the concept of non-dominated set is very important in the concept of optimization of objective function.

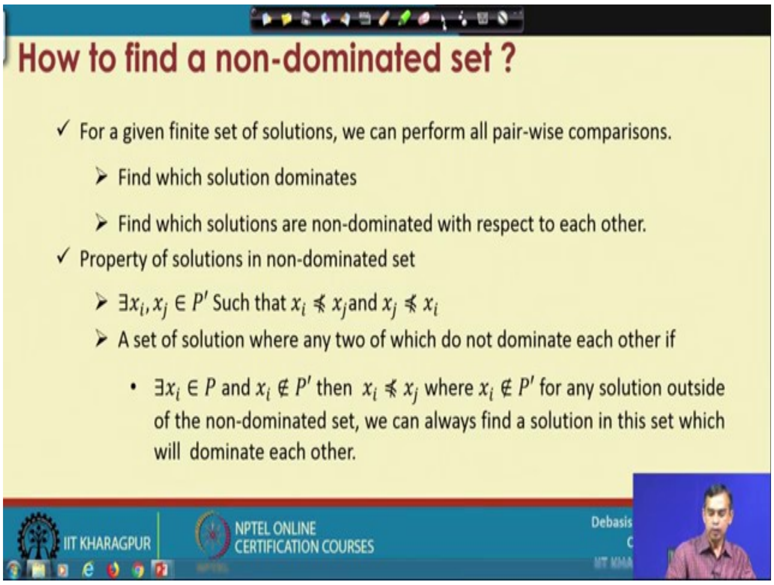
(Refer Slide Time: 10:37)



Now, again we can elaborate it, so the same idea that we have discussed about. So, non-dominate in this case is 3 and 5; that means, P' and P is the all solutions in the solution space.

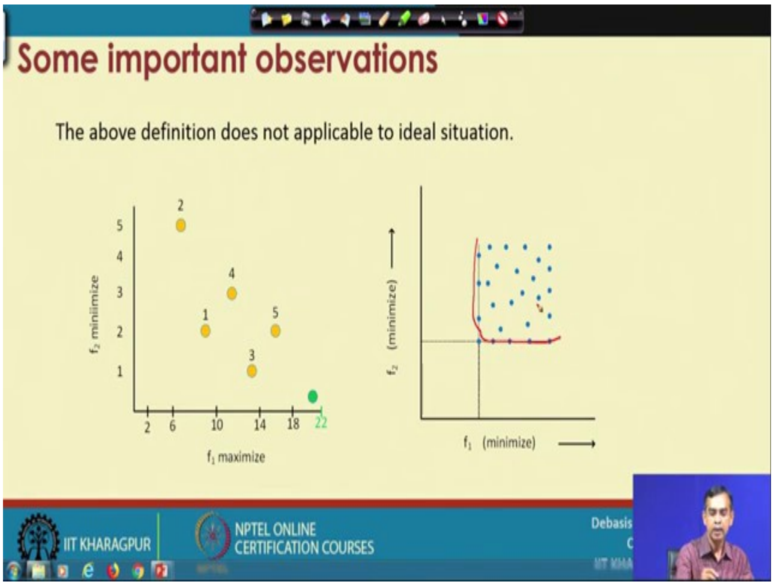
Now, if it all solutions this all solution in the all solution, then this is the optimal front and particularly this optimal front is called a pareto optimal front.

(Refer Slide Time: 11:12)



Now, we will discuss few cases are there so that we can learn about how to find a non-dominated set ok. So, basically we have to apply the dominant solution for all solution with respect to any other solutions and if we can find the solution which is not dominated by any other solutions, but ok. So, then it is the non-dominated front like this.

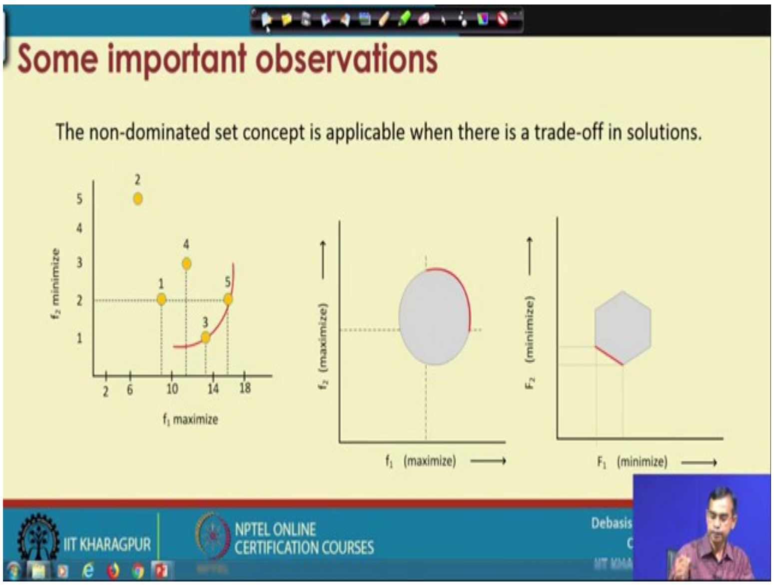
(Refer Slide Time: 11:44)



Now, ok, so this is an idea that the solution can be obtained by this concept of dominant solution and then we can find it. Now, so, again we can note that if there is an ideal, there is an ideal of solution, so ideal solution does not satisfy these kind of concept actually. So, it is not applicable to, so there is no front actually.

So, for example, if this is a solution space both f_1 is minimize and f_2 minimize then, then in fact, this is not the front actually rather the ideal objective front only one solution. So, this is not the front actually. So, this is the only solution that is there. So, for the ideal solution is concerned.

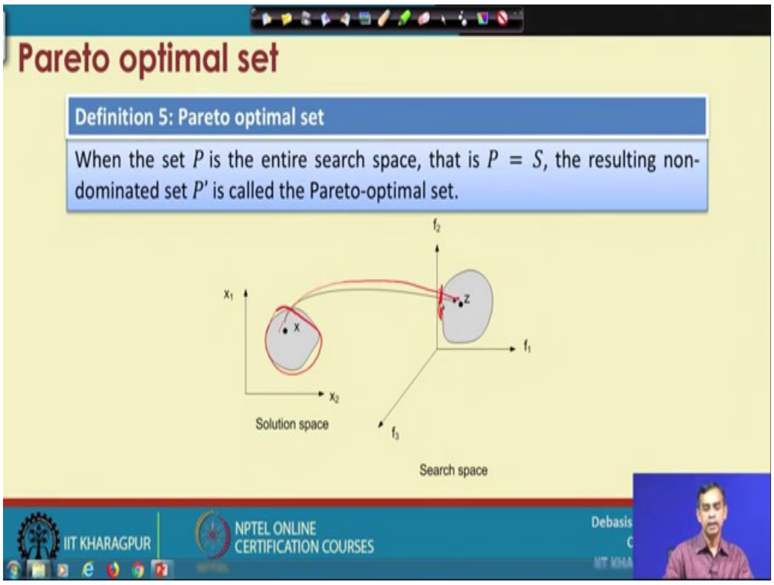
(Refer Slide Time: 12:41)



Now, we can generalize this concept with reference to many examples if it is there. Now, if you can say that if it is maximized if it is minimized as we have learned about this is the pareto optimal front. Now, if we consider another case f_2 is maximise and f_1 is maximise again then the this is the front and this is the entire solution out of this entire solution this is the front; that means, all the solutions which lies on this region they are called the pareto optimal solution.

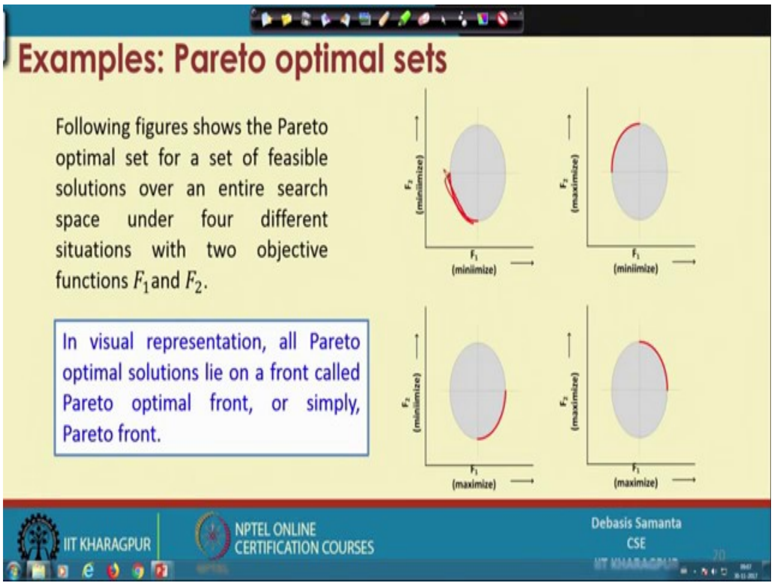
An another solution if f_1 is minimized, and f_2 is minimized then these are the solutions are the pareto optimal solution. Then this front is called as pareto optimal front because its satisfy this concept of domination.

(Refer Slide Time: 13:46)



Likewise, you can extend few more example ok. So, we have learned about the pareto optimal set. So, if the, so pareto optimal set this is the entire solutions space, out of this entire solution space which are on the pareto optimal front then they are the pareto optimal solution. This is the concept there.

(Refer Slide Time: 14:14)

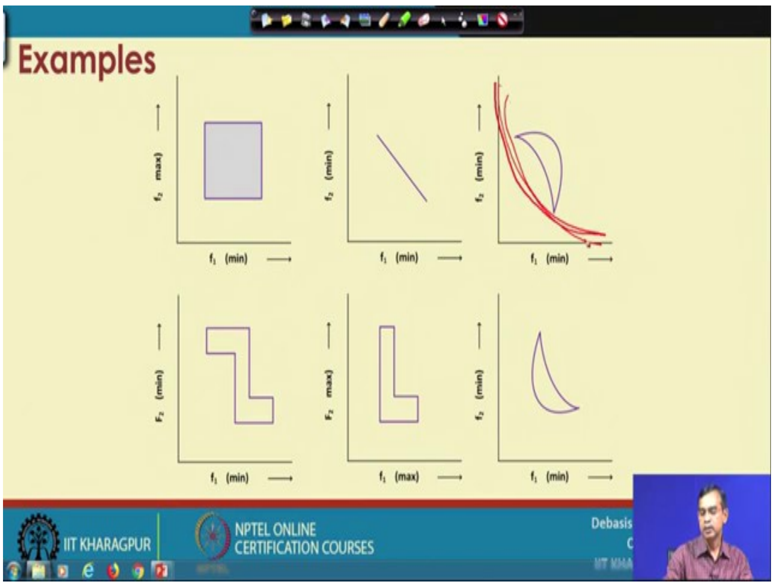


Now, few examples in order to identify the different concept of the different situations where the pareto optimal solutions because some time we have to decide only the; so it is the pictorial description of the different situation by the pareto optimal solution can be thought of. Now, here, depending on f_1 is minimise and f_2 is minimise, so this is the pareto optimal front. Similarly, if it is maximise and if it is minimise then this a pareto optimal front and if it minimise and it maximise this is the pareto optimal front and it is maximise and maximise this is the pareto optimal front and these are the entire solutions space or the search space, the entire search space that is there.

So, these is the concept here. But we should not worry about the different situations there it is just only matter of understanding the important thing is that any objective function whether it is minimise or maximise or whatever it is there they can be converted into one form, either all maximise or all minimise then our idea will be very simply. If we all minimise then we can say that these are the concept. So, optimal solution we can easily identify a particular front which basically the pareto optimal front. So, the idea is that all the solution is given to you and as I told you the pareto optimal front is our desirable solutions. So, if we can identify the pareto optimal front then we can take all solutions and these are the trade-off solutions.

So, all pareto optimal solutions which lies on a front is called the pareto optimal front sometimes it is simply called as the pareto front. Now, this is the idea that we have illustrated for the two objective function is very difficult to visualize in case of n dimensional, if there n objective vector is there, but it is the concept a mathematically the same concept can be applied whether the two objectives or more than two objectives are there.

(Refer Slide Time: 16:21)

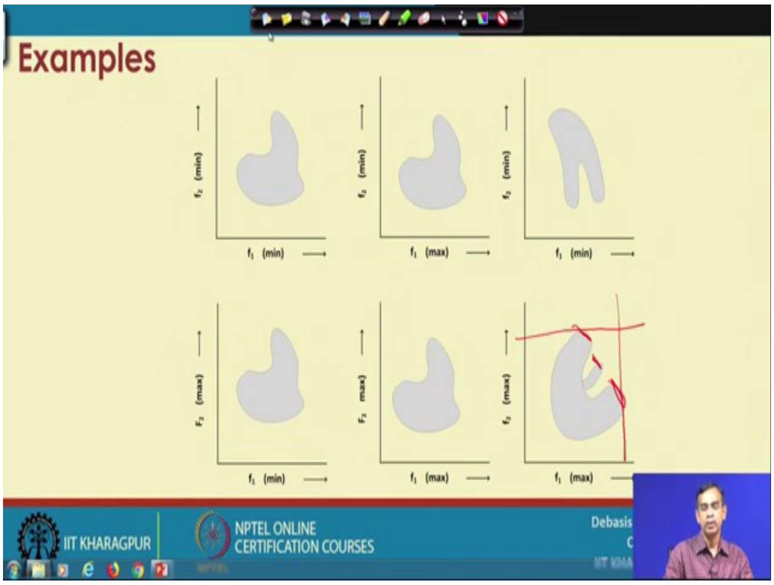


So, this is the concept about it and we would conclude this concept with the few examples here. Now, the first is that the $f(min)$ and $f(max)$. So, it is a basically maximising this one and minimising this one. Now, you can say that which is the pareto optimal front here in this case. So, in this case because if it is maximise and it is minimise we can re call in the one slides in the last lecture we have different the cases there we can say that this is the pareto optimal front in this case.

Now, second example here both f 1 and f 2 are minimise. So, this is the pareto optimal front in this case. Now, here is a one another the typical curve it is here. So, f 1 is minimise f 2 is minimise. So, this is the entire search space then this is the one front this front is basically pareto optimal front; no, no; so is ok. So, is basically ok. So, if it is minimisation, so now can you tell me which is the pareto optimal front. So, basically it is basically so far the minimisation is concerned this is a function which basically the able the basically the pareto front look like the. So, here in this case this one and this one are the two solutions which are lies on the pareto optimal front.

Now, here if it is f_2 is minimum and f_1 is minimum both the minimisation and this is the front then then we can say that these are the pareto optimal front in this case. Now, similarly for the max if it is here. So, then these and these are the pareto optimal front in this case. And here also min and min and this is the pareto optimal front in this case. So, you have, so given the different what is called as geometry of the solution space we will be able to find what are the different front there and which front is essentially the pareto optimal front that is an important thing, that you should learn it and you should know it.

(Refer Slide Time: 18:41)



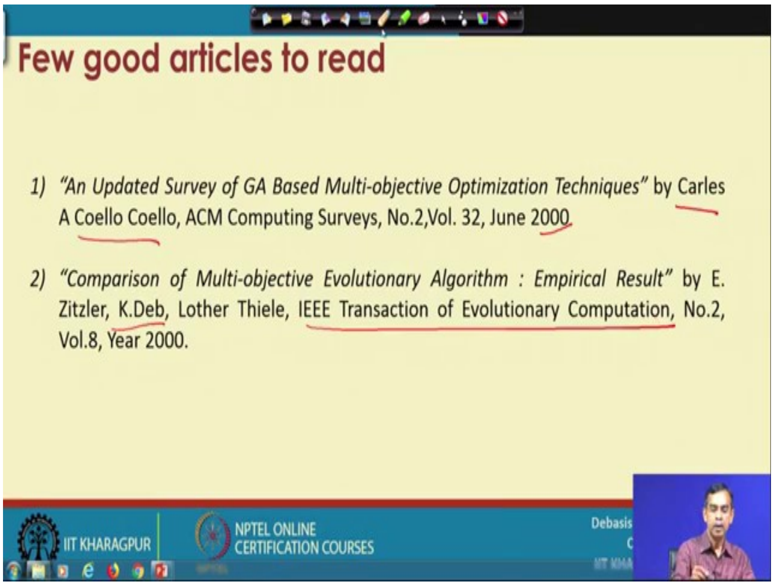
After visiting few examples I would like to give few more examples actually it happens in many real life solutions are there ok. Now, I left is an exercise for you. So, you can check it and then verify it. Now, here a f_1 is minimisation and f_2 is minimisation then you can find which is a pareto optimal front. So, it is basically it is like this and I just given an example for your hint. So, this region is the pareto optimal front in this case. Now, likewise the same idea can be explained if it is maximise and it is minimize then this front is the pareto optimal front in this case.

Now, here again minimise. So, it is basically minimise and this is minimise. So, this and this, so these are the pareto front and this is a pareto front in this case. Now, again it is minimise and it is maximise, so it is minimise and maximise. So, this concept it is like this. So, minimise and maximise; so it is ok. So, it is maximise means this one and minimise this one. So, this front is basically the pareto optimal front in this case. Now, is a maximum, is a maximize.

So, it is basically maximize and this is the maximize, so this is pareto a front in this case. So, the pareto front like here and for the maximum this is basically maximum and this is maximum. So, this front is basically a pareto front in this case. So, pareto front not necessarily be a continuous front actually it may be discrete front as we have illustrated with few examples here.

So, we have learned about the different solutions and then we discussed about the ideal solutions. We discussed about the utopian solutions and their application. And then we have discussed about the concept of domination and then the relation that the domination can satisfy, and then we have discussed about the pareto optimal front which is important to understand about different solving the multi objective optimization problem in this case.

(Refer Slide Time: 21:08)



Now, so, in order to understand this concept better I would suggest to follow few articles because it needs lot of patience and more studies to understand the concept. So, the fast there is a survey paper it is an updated survey of GA Based Multi-objective Optimization Techniques by Carles A Coello and Coello this is and this is published by ACM Computing Surveys in 2000. So, this is very good article to read.

And there is one very nice paper which is written by K. Deb, Kalyanmoy Deb. He has many contributions in the field of multi objective optimization solving and here basically comparison of multi objective evolutionary algorithm some results by Zitzler, Deb and Thiele. It is publishing IEEE transaction of evolutionary computation. Now, this is a one transactions published by IEEE is very famous and many articles related to our discussion can be obtained from there. So, these are the paper that you can follow to understand the concept.

So, with this discussion I would like to stop it here. And we will discuss about the approaches, the different approaches to solve multi objective optimization problem that we will start in the next lecture.

Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

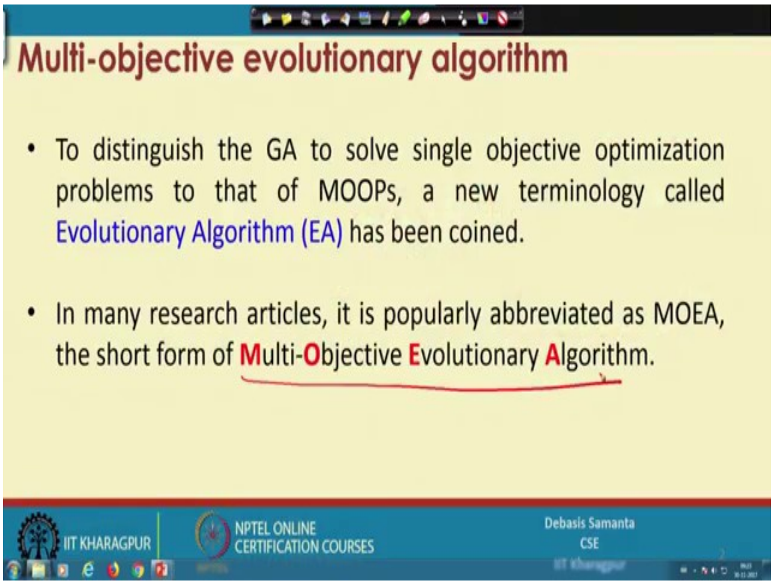
Lecture – 27
Non-Pareto based approaches to solve MOOPS

We have learned some concepts regarding the multi objective optimisation problem. Now it is a right time to discuss about the different approaches. In the last two decades, there is a huge research is to develop the best solution solving multi objective optimisation problem and then it opens up further many other what is called the areas. So, that further research can be extended and as the number of approaches are very large.

So, it is not possible to discuss all the approaches out of the different approaches we will try to discuss the popular and then path breaking approaches or we can say they are the state of the art approaches to solve multi objective optimisation problem. Now, we will discuss the different approaches in a when they are discussed in a different chromogical sense; that means, the first approaches we will be discussed here we will be discussed first here and then the latest the approaches in this field will be discussed at the end.

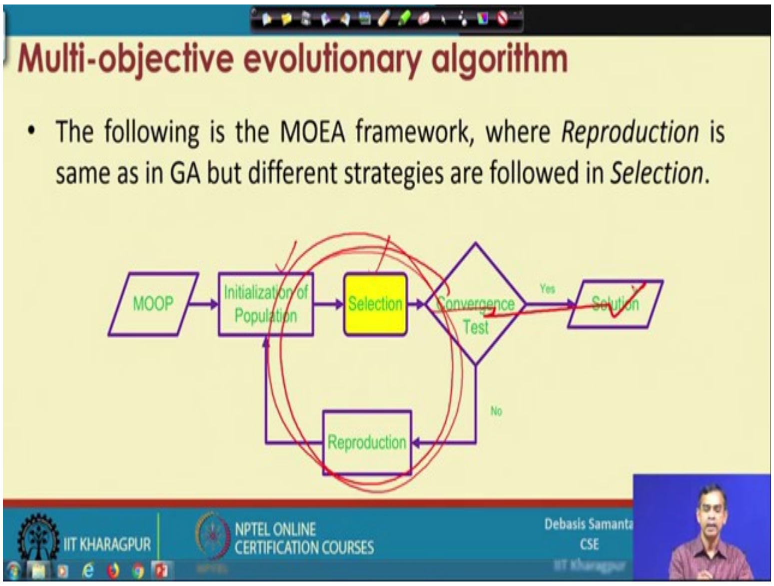
So, it will take few classes of course, to cover all the approaches, now we will discuss about multi objective optimisation problem solving approach in order to make a difference between the traditional Genetic Algorithm to solve single objective optimisation solving, but using the same as a GA concept then this technique is also alternatively called MOEA. So, it is MO stand for Multi - Objective and then EA it is called the Evolutionary Algorithms. So, algorithms are particularly termed as evolutionary algorithm. So, it is more popularly termed as MOEA algorithms.

(Refer Slide Time: 02:12)



So, we are going to discuss about the different MOEA algorithms to solve multi objective optimisation problem.

(Refer Slide Time: 02:28)



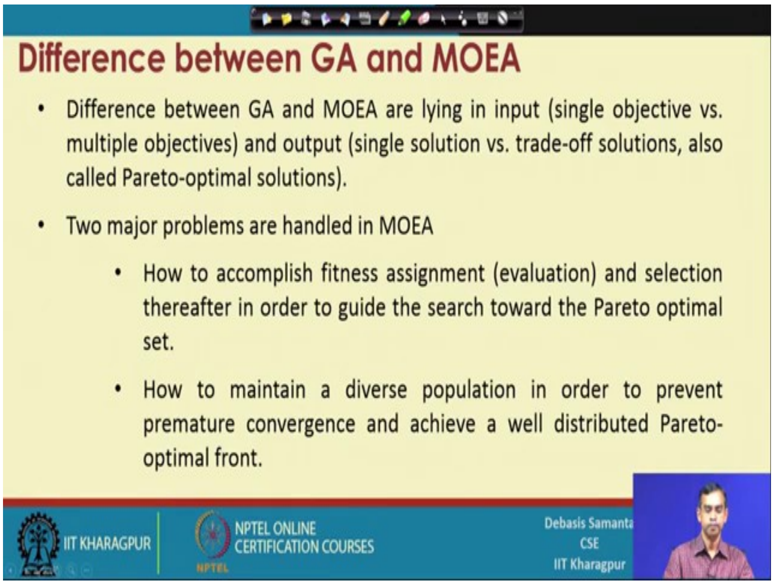
So, for the concept now we are going to discuss the concept is that using genetic algorithm framework that we have learned earlier to solve the single objective optimisation problem then how they can be used to solve multi objective optimisation problem and you can recall we are discussing about two approaches one is called the a priori approaches and then a posterior approaches.

So, it is the same concept it is here also we will discuss about on the top of a priori and a posterior and they can be further classify as a Non-Pareto based approach and then Pareto based approach we will discuss the classification shortly. Now let us first start to it how the simple genetic algorithm framework that is the GA framework can be applied. So, initially the attempt was to apply the GA framework to solve the multi-objective optimisation problem that is why it is called the GA framework to solve MOEA problem or it is also called MOEA framework.

Essentially the MOEA framework and GA framework are same because in case of GA framework you can understand. So, you have to create the initial popularization and then the selection and then we have to do the convergence test and then the reproduction. So, it is basically repeats of selection and reproduction until this convergence test is satisfied to find the solution. So, these are same as the GA framework here, but there is a difference the difference in the sense that ok.

So, far both GA and then MOEA framework is concern these steps are same, all the step remain same; however, the only step which is different is called the selection. This means that to solve the simple of single objective problem the selection method is followed that we have discussed when we are discussing about solving single objective optimisation problem using GA framework, but here in case of solving the multi - objective optimisation problem a totally different selection method methodology are followed and this way the GA framework is different or a MOEA framework is different than the GA framework. Now we learn about what exactly the selection strategy that is followed in MOEA and that is the different selection means is a different MOEA algorithm in fact.

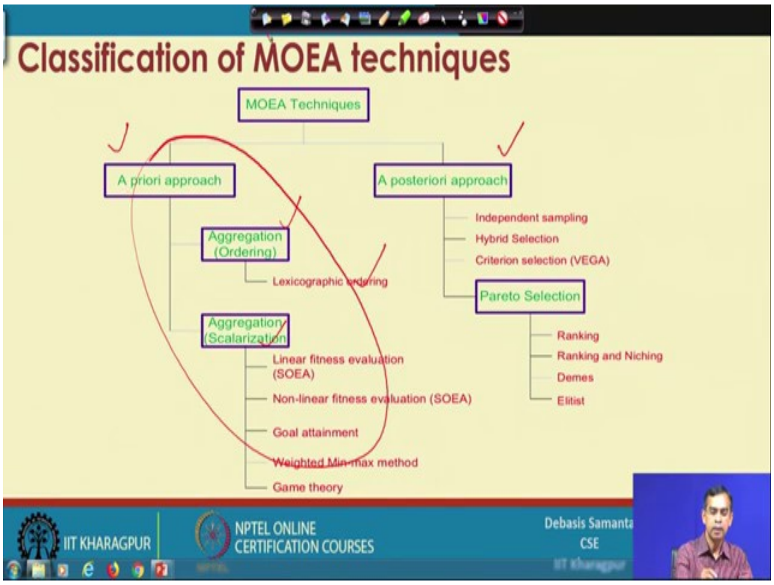
(Refer Slide Time: 04:58)



So, we have learned about the difference between the GA and MOEA. So, they are in the sense that GA is useful for solving single optimisation problems whereas, MOEA framework is used for solving multiple objective optimisation problem. Now so, far the input is concerned they have the same input; however, so far the output is concerned as you know single objective optimisation problem is basically gives the single solution. However, for the multi multiple objective problem they are not the single solution rather trade off solution and they are also called the Pareto optimal solution.

Now, so far the different issues are concerned in case of MOEA approach so, basically the idea is that MOEA the fitness assignment that is the evaluation then followed by the selection needs to be considered in a different manner than the single objective optimisation problem and the another issues so, far the MOEA framework is concerned is that how we can maintain the diversity in the population. So, that we can search towards the Pareto optimal front only. That means, we have to direct our searching procedure. So, that out of the entire surface the solution can leads to the Pareto optimal front. So, if we can find the Pareto optimal front and that is basically the trade-off solutions to solve the multi - objective optimisation problem.

(Refer Slide Time: 06:40)



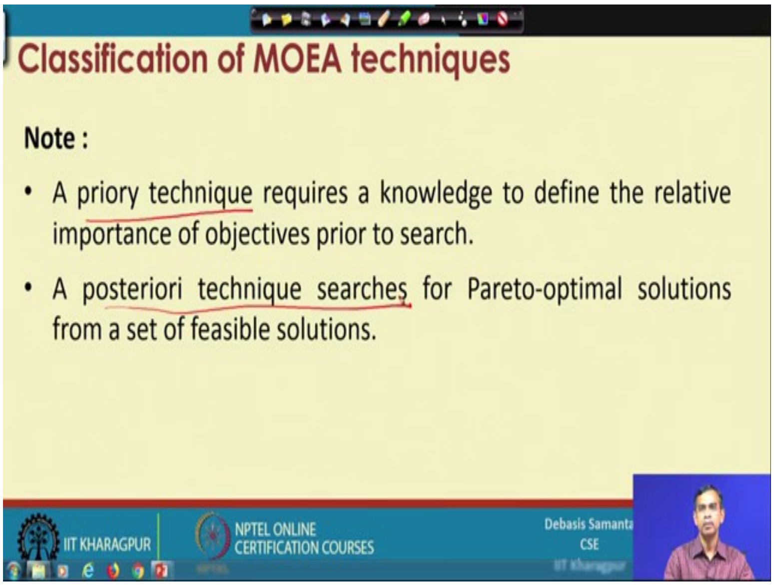
So, this is the concept between the GA this is the difference between the GA and MOEA techniques. Now, let's see what are the different approaches are there, whether I should discuss about the taxonomy of different MOEA techniques now. So, all the MOEA techniques can be broadly classified into two broad categories they are called a priori approach and another is called a posterior approach.

So, for the a priori approach is concerned again there are different division one is called the a priori approach based on aggregation or ordering and then another is based on scalarization. So, an example of a priori approach is ordering is called the lexicographic ordering and there are many other methods following the game theory, weighted min -max method, goal attainment method, non-linear fitness evaluation scope it is called the SOEA linear fitness evaluation it is also called SOEA.

So, these are the different methods are there so, for the a priori approach is concerned. So, this is the a priori approach approaches and then a-posteriori approaches they can be of different types one is independent sampling or hybrid selection and then vector evaluated genetic algorithm it is also called VEGA and there are many other methods again a posteriori approach ranking, niching, elitist and demes, so these are the different methods are there.

So, these are the techniques as you know we have considered we have listed here only few because we cannot list all the solutions here, but these are the state of the art solution. So, for the different MOEA techniques is concerned all the solutions have their own prose and own cons that will be discussed while discuss it is topics, it is techniques individually. So, these are the different techniques are there and all these techniques again there is a different classification also there.

(Refer Slide Time: 08:48)

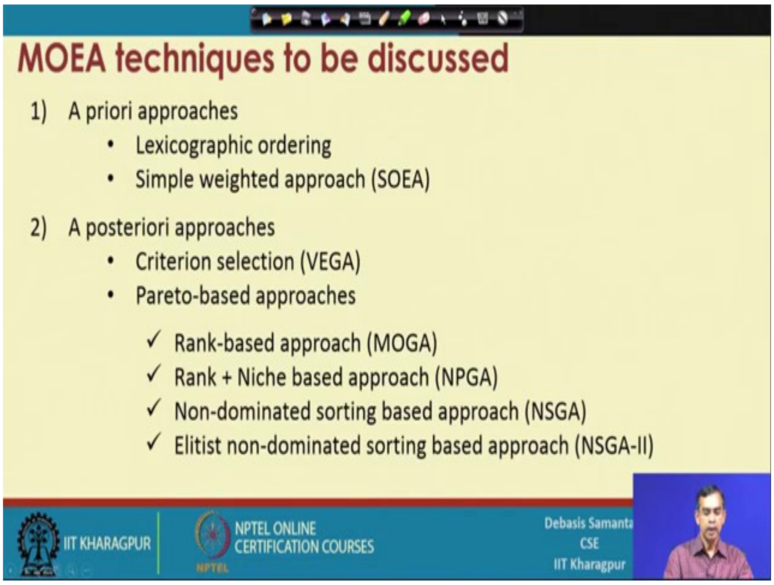


Now, a priori techniques as we have mentioned there is that it requires a knowledge to defined the relative importance of the objective vectors prior to the search then it is called the a priori approach. On the other hand, in case of a posteriori technique it does not required any prior knowledge, it basically required certain techniques or some mechanism by which it can limit the search towards the Pareto optimal front only.

So, this is the concept that is there so; obviously, the first approach; that means, a priory techniques required a lot of experience for the programmer whereas, this is computationally expensive; however, it does not require any knowledge about the solving problems or the different strategies to be adopted in order to apply a particular technique is there.

So, these are the I means major what is called the pros and cons the a priori approach is first whereas, a posteriori technique is computationally expensive, but a priori approach requires the knowledge of the programmer where a priori technique does not required the any knowledge of the programmer. So, these are the merits and demerits in the two techniques there.

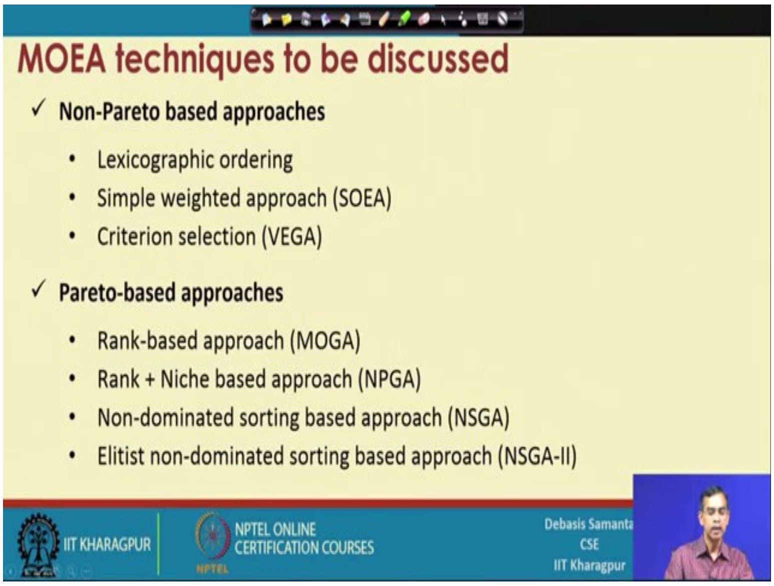
(Refer Slide Time: 10:10)



Now, again so, we have discussed about all the MOEA techniques based on a priori approach and a posteriori approach as we have listed all the techniques, which are belongs to a priori approach and a posteriori approach the different techniques that will be discussed in this course is basically so, far the a priori approach is concerned we will discuss two techniques not all techniques; however, so, these on.

So, for the a posterior approach is concerned we will the discuss these are the techniques will be one by one and again all these techniques whatever the techniques all these techniques again can be classified into different whether they are Pareto based or Non-Pareto based.

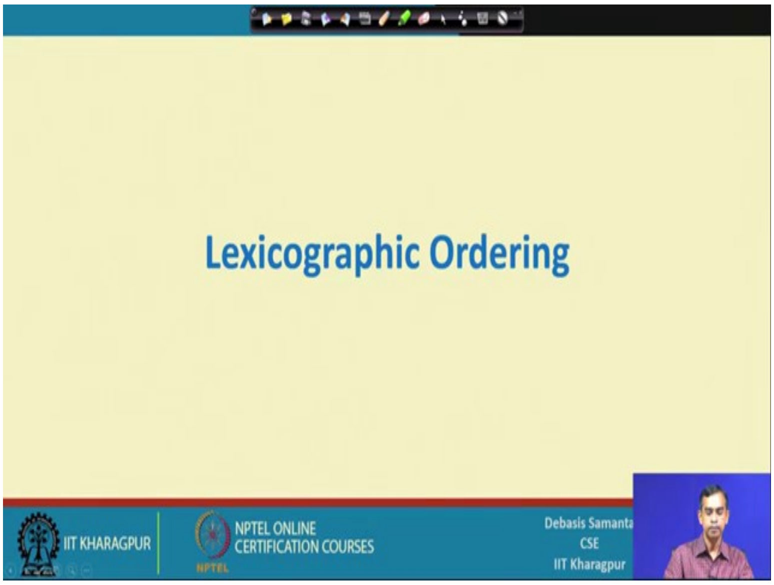
(Refer Slide Time: 11:00)



So, here is another division based on the Pareto based approach and Non-Pareto based approach. All the techniques these are included here are the Non-Pareto based approaches and these are the Pareto based approaches.

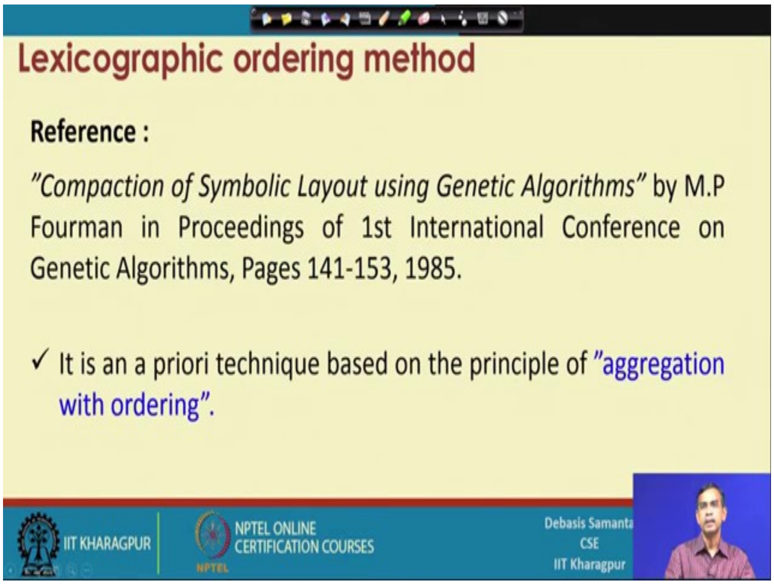
So, this way the different techniques can be classified which belongs to a particular concept or principle Non-Pareto based, then Pareto based. So, first we will start about discussing about the Non-Pareto based approaches and we will discuss the first approaches in this direction is called the Lexicographic ordering.

(Refer Slide Time: 11:33)



Now, it is an a priori approach as well because it requires the knowledge of the ordering of the objective vectors.

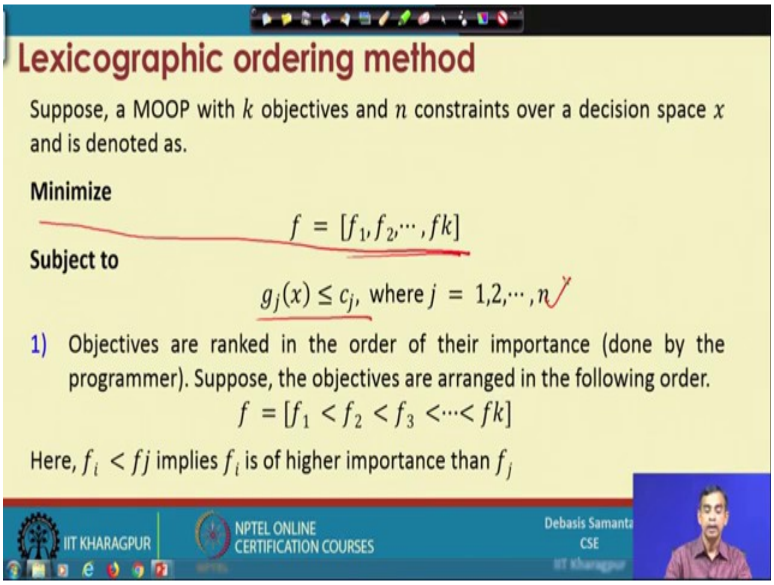
(Refer Slide Time: 11:43)



That means there are if m f n number of objective functions are there you should have a knowledge about that which objective functions are most important than the others one or like this or some relative ordering with respect to their evaluation or their importance. Now the approach the lexicographic approach first time proposed in 1985 by Fourman he introduced this concept and published a paper entitled as compaction of symbolic layout using genetic algorithm it was published first time in 1985 in the conference first international conference on genetic algorithm.

And it is a priori technique as I told you and the principle it follows is based on the aggregation with ordering now we will understand how why this principle is said so, aggregation and what is the concept of ordering, now let us discuss this concept.

(Refer Slide Time: 12:45)



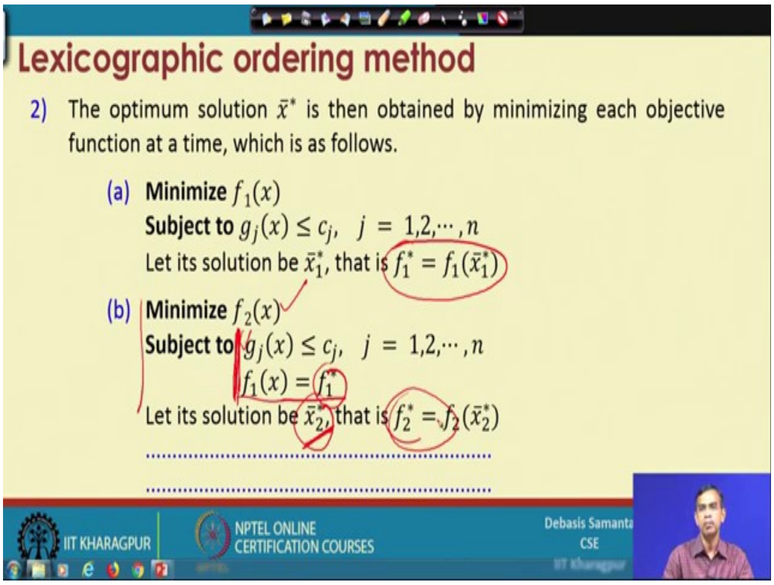
And without any loss of generality again I should say that there are k number of objective functions in this concept. So, k number of objectives can be denoted as $f_1, f_2, \ldots, f_k$ and we will assume that all objective functions are to be minimise, this is not a contradiction or any problem because all the objective function can be converted into minimisation type also. So, we will consider that these are all the objectives are to be minimised and like all objective of that for any objective problem optimisation problem there are the constant.

So, let these are the constant, constant is denoted i . So, n number of constants are considered there and as I told you this lexicographic ordering technique required ordering of the objective vectors. So, we will consider the one objective ordering of the objective vectors are like this. So, far the importance is concerned that in this order then f_1 is first then, f_2 is then, f_3 is then, f_k is there.

So, this ordering like this f_1 is most important, f_2 is less important, then $f_3 < f_2$ and so on. So, it is basically here we can say that $f_i < f_j$ implies that f_i is of higher important than f_j this is the concept that let us assume it and then. So, this ordering is known a priori. So, this ordering is known a priori then we can follow the idea about lexicographic ordering.

Now so, the idea or the step that is followed in lexicographic ordering is first we have to rank all the objective vectors into their importance of an importance; that means, ordering there. So, this is the first step that we have to follow it ordering is the first step once the ordering is done then our next step is a basically iterative steps.

(Refer Slide Time: 14:57)



So, the first is that we have to consider only one objective function in their order. So, in this order f_1 comes first. So, we have to first minimise only $f_1(x)$ without considering the other objective functions and with the same subject the constant it is there n number of constants.

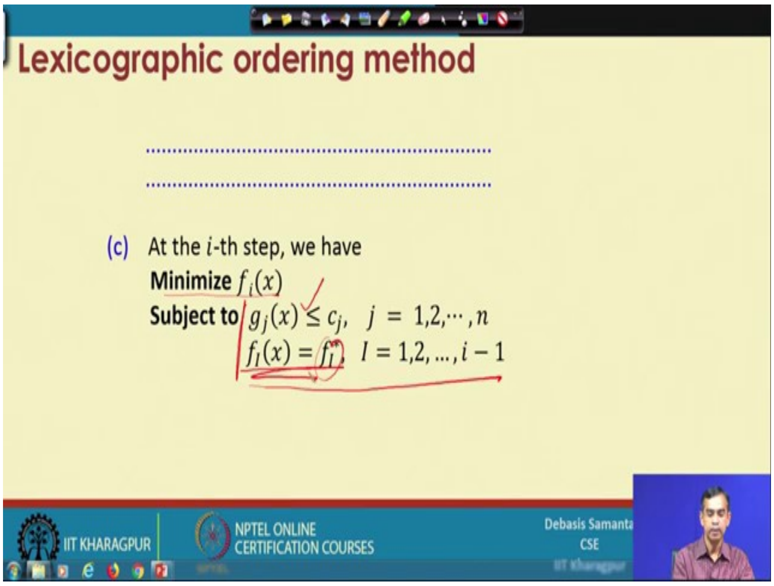
So, this can be considered as the simple or single genetic single objective optimisation problem solving and for this we can follow the simple GA framework in fact. So, these suppose so, these leads to so, when we apply using single objective optimisation problem using GA framework and then it will give solution let the solution in case gives $\acute{x}_1^{¿}$. So, $\acute{x}_1^{¿}$ is an optimum solution with respect to the first objective f_1 .

Now, the second step we have to then find solution objecting the second objective function. So, it is in this case minimise $f_2(x)$, but the constant will be whatever the other constant it is there it is the original problem and another constant that $f_1(x)$ $¿f_1^{¿}$, where $f_1^{¿}$ is basically the solution that is often from the first step.

So, on constant will be included here in order to solve the second ranked objective function namely f_2 there and this will give a solution say $\acute{x}_2^{¿}$; that means, with respect to only optimising $f_2(x)$, $\acute{x}_2^{¿}$ is the solution that is the optimum solution to with reference to $f_2(x)$ and with respect to these are the constant and let this be $f_2^{¿}$. So, it is basically optimum we with respect to second objective f_2 .

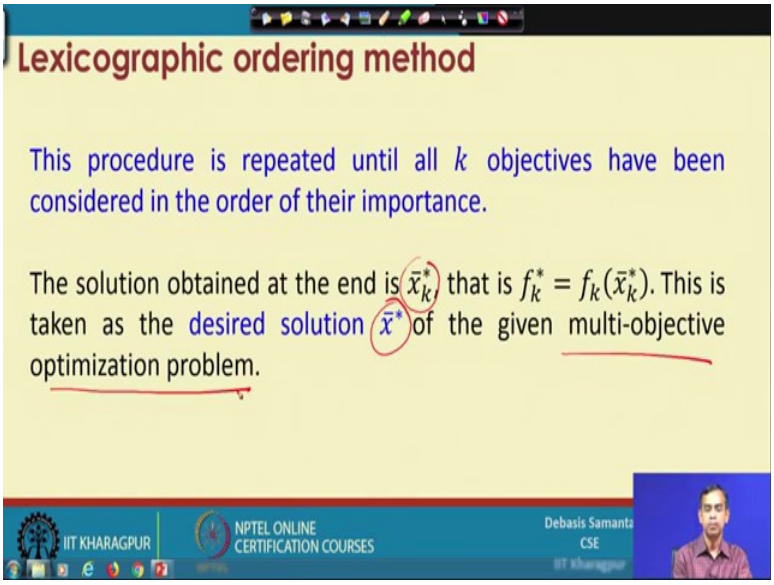
Now, we will repeat the same thing again, but for the next ranked objective function like f_3 and f_4 and so on.

(Refer Slide Time: 17:03)



So, finally, at any i - th iteration. So, the objective is to find a solution targeting the minimisation of the i - th objective function $f_i(x)$ and the constant will be again increases one by one. So, it is the original constant for the all objective function and they are will be for the i - th iteration $(i-1)$ number of constant will be added which is basically the solutions that we have obtained with respect to $f_1, f_2, \ldots, f_{i-1}$. So, these are the solutions and it satisfy this condition. So, this is the idea that is followed there.

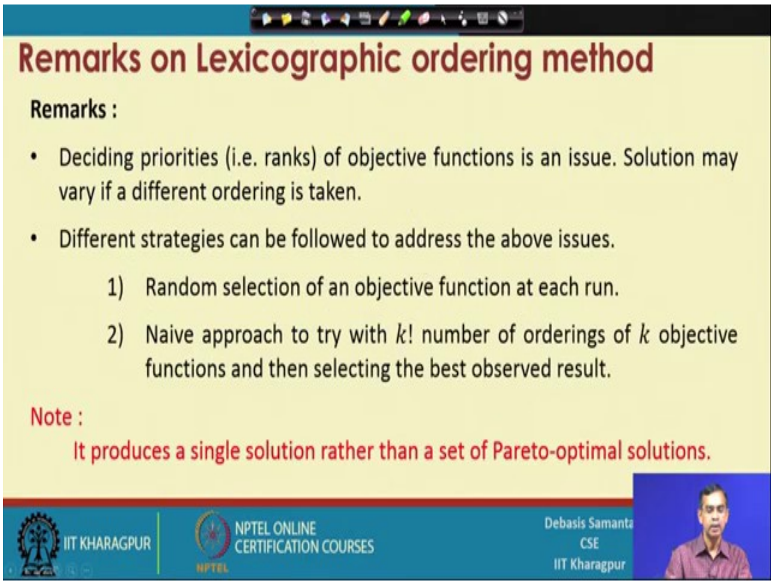
(Refer Slide Time: 17:53)



So, obviously, after the k iterations, we will find a solution let this be the solution is let this be the solution $\acute{x}_k^{¿}$ right that is f_k is basically solving the k th optimisation problem at the end and when the constant is all the constant that is there in the original problem plus the constant $f_1(x)$ ¿$f_1^{¿}$, $f_2(x)$ ¿$f_2^{¿}$, … $f_{k-1}(x)$ ¿$f_{k-1}^{¿}$ this one. So, this solution after the, this one is basically the desirable solution and we can term as this solution as the optimum solution.

Now, this is the solution that can be obtain after the repetition of the k objective in succession one by one according to their order of importance and this way the solution can be obtained as a multi objective optimisation problem and you can understand that. So, here the solution that can be returned by this approach lexicographic ordering is only one solution instead of many solutions that is called the trade-off solutions there that is why it is also called Non-Pareto because it does not give any Pareto front or Pareto optimum solution it is an a priori approach as well as it is a Non-Pareto based approach.

(Refer Slide Time: 19:22)

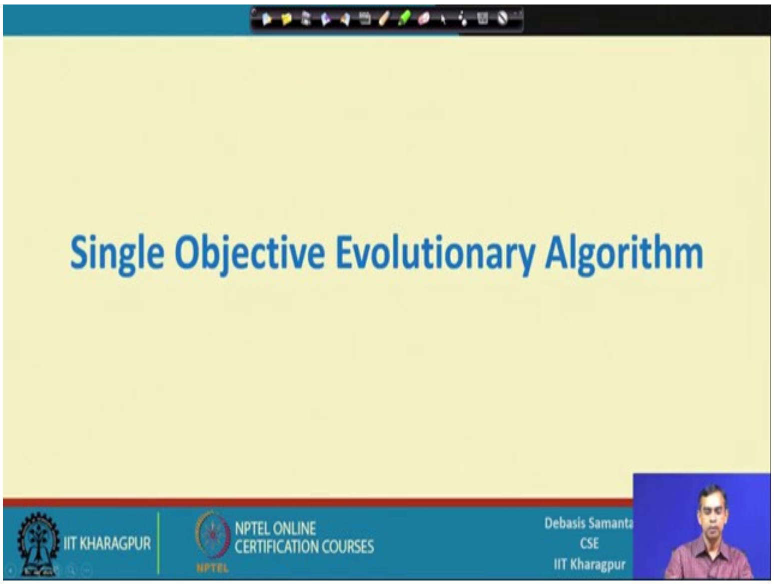


So, this is the concept of lexicographic ordering and; obviously, there are certain criticisms about this one. So, the first criticism regarding this method is that we have to decide the priorities of all the objective functions that is there in the multiple objective optimisation problem. That is we should have a correct or the knowledge of correct or actual knowledge of the ranked of the different solution.

If we do not have any ordering of the concept of or knowledge on ordering of the any objective function then it may leads to non-erroneous result or non or you can say that non - optimal solution. Now; however, if you do not know the orderings of the solutions then there may be some strategies can be found out. One is that random selection of objective function at each run, but you know there are say if k number of objective is there then finding random out of which the best is a basically is a $k!$ search.

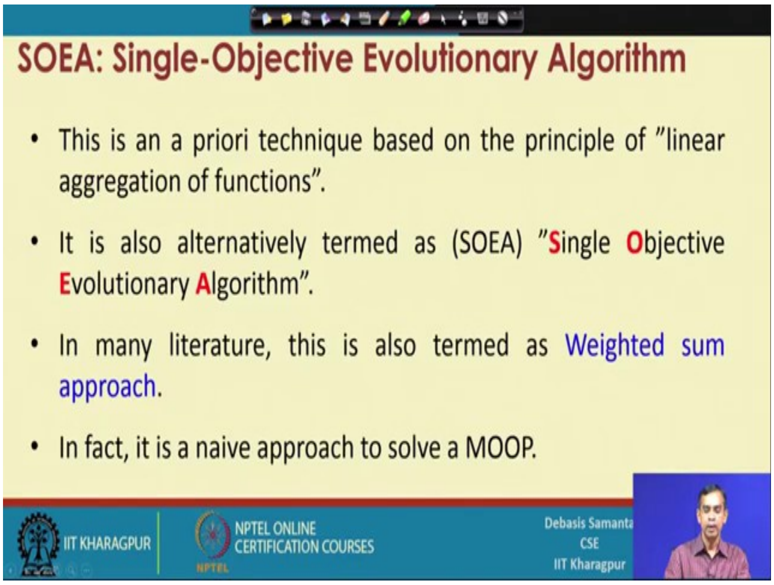
If the number of objective factor is only 2, 3, 4 it is possible to apply this one, but if the number of objective factor is very large then this method is cannot be applied. So, this is really one serious drawback of this of this method and as I told you if it is an objectives are in conflictive nature usually we have to find a trade of solutions that is a Pareto optimal solution, but lexicographic order gives a single solution instead of trade off solution that is more desirable than the multi - objective optimisation problem solving is concerned.

(Refer Slide Time: 21:04)



So this is the lexicographic ordering next another simplest method which also can exercised using the same as a GA framework it is called the single objective evolutionary algorithm. Now it is basically the idea is that if there are multiple objectives how we can convert this problem into a single objective optimization problem. So, this is the basic idea that is followed here in this technique and this technique is also called SOEA it is Single Objective Evolutionary Algorithm (SOEA). So, it is SOEA approach is there.

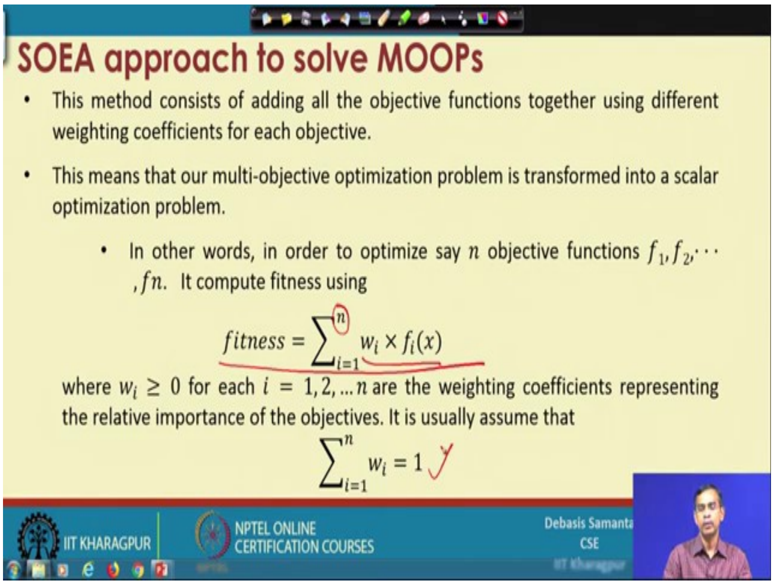
(Refer Slide Time: 21:40)



Now, so, it is also alternatively called Weighted sum approach because it basically considers some weights in order to find the in order convert the multiple objective optimization problem into a single objective optimization problem. It is on basically nascent approach published as early as the lexicographic ordering solution was proposed in 1985 or so.

Now, let us see exactly, what is the technique that is followed there in case of SOEA approach.

(Refer Slide Time: 22:08)



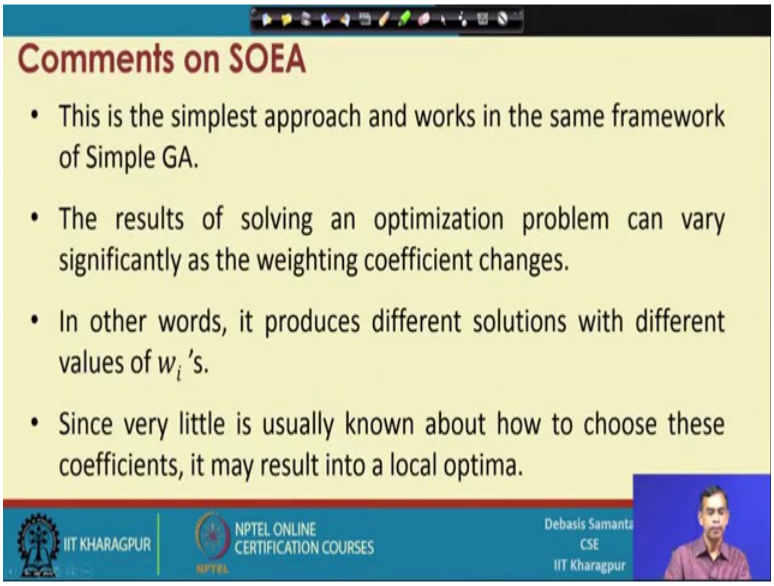
First we have to decide the weighting coefficients for each objective function. So, in case of lexicographic ordering it requires to know a prior knowledge about their ordering of the objecting function. Here is a same way of course, it is not exactly the ordering rather weights of the objective function. In other word, if suppose f_1 is most important then we should a weight w_1 which has higher value than the least important less important objective functions say f_2 and then weight is w_2.

So, we have to decide if there are k number of objective functions the k number of weight coefficients weighting coefficients $w_1, w_2, \ldots, w_k$ and once the k number of weighting coefficients are known to us then we will be able to formulate a fitness value; that means, the single objective there. So, this within this formula. If the n objective functions are there so, is basically sum of products of weights and then their objective values.

So, it the w_i and then $f_i(x)$ and summation of all these and there. However, all the weights are to be decide in such a way that the $summation\, of\, the\, weighting\, coefficients = 1$. So, this is a normalized form of the weighting factors weighting values and. So, here the main important most concern about how to decide w_i. So, once it is decided correctly we will be able to get the solution correctly.

So, this is the idea about the single objective evolutionary algorithm or it is called the SOEA approach to solve multiple objective optimization problem and as this is a single objective optimization problem like this here; that means, the same genetic algorithm framework can be applied there without any change, basically the same reproduction, the same selection, the same initial population creation, cross over mechanism and convergences all these things can be applied here.

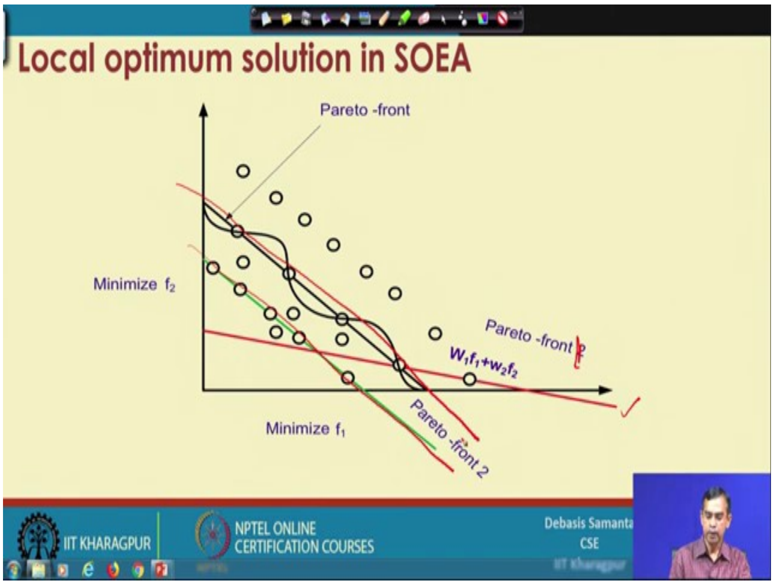
(Refer Slide Time: 24:33)



So, this is the idea about SOEA approaches and definitely SOEA approaches one is a simple most approaches because it can be used, it can be solved using the simple GA framework; however, the results of solving an optimization problem can vary significantly if their weighting coefficients values are changes; that means, for different a different solutions can be obtained for the different values of w_i. So, that is why all the weighting coefficient values are to be decided as accurate as possible to get the best solution.

So, since very little information is known to us how to choose this coefficient usually this solution may leads to a non-local solution or is a non- global solution is a local optimum solution rather. So, usually so, it is simple, but solution may not be. So, accurate it is very fast compared to other MOEA techniques there.

(Refer Slide Time: 25:43)



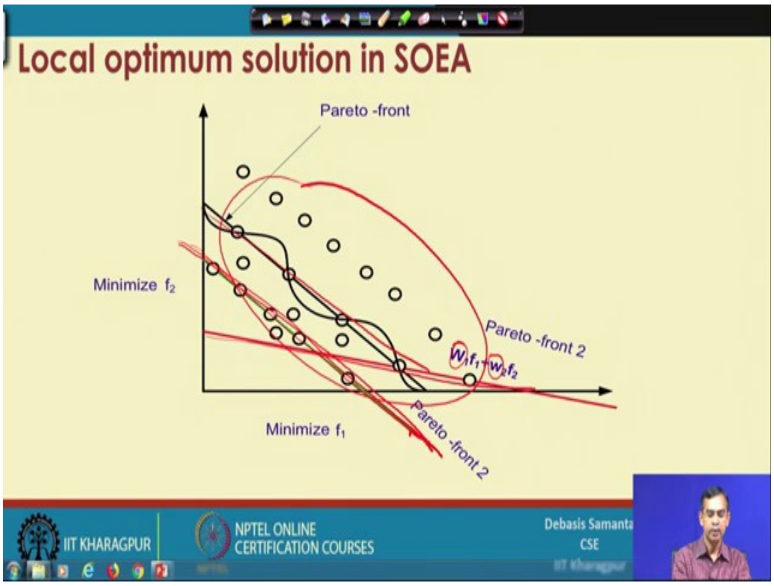
Now, just let me illustrate about how if you decide the different weighting factors and then it gives to the different solutions actually. Now here the idea it is there, so the vector or you can say single objectives that can be represented by this one. So, it is basically single objective this one and if w_1 and w_2 . are the weighting coefficient and this is basically f_1 and f_2 . So, it can be plotted at the linear function of $w_1 f_1 + w_2 f_2$ for the unknown f_1 and f_2 .

So, it is just like a and you know. So, $w_1 f_1 + w_2 f_2$ in a 2 Dimensional space f_1 and f_2 it basically represents a straight line. So, the idea it is there so, for different w_1 and w_2 values, with two objectives f_1 and f_2 the different solutions can be obtained. Now each solution in basically, so if this the one line corresponding to one value $w_1 f_1 + w_2 f_2$. Then it basically gives one front like. So, this can be termed as a Pareto optimal front 1.

So, all the solutions which lies on this is basically the trade-off solutions. So, for our this is concerned and definitely it will return only one value depending on the values of f_1 and f_2 . So, suppose either it written this one or it written this one depending on a f_1 and f_2 and depending on f_1, f_2 or if any other solutions are there. So, essentially it try to find a Pareto optimal front, but actually it returns one single solution depending on the f_1 and f_2 .

Now again this Pareto optimal front that we have that we can obtain using single objective evolutionary algorithm varies if the coefficient values are varies. For example, so, if this the Pareto optimal front 1 then another for the different values of w_1 and w_2 another Pareto optimal front can be obtained or this is another Pareto optimal front can be obtained for the different coefficient values of f_1 and f_2 different coefficient values of w_1 and w_2.

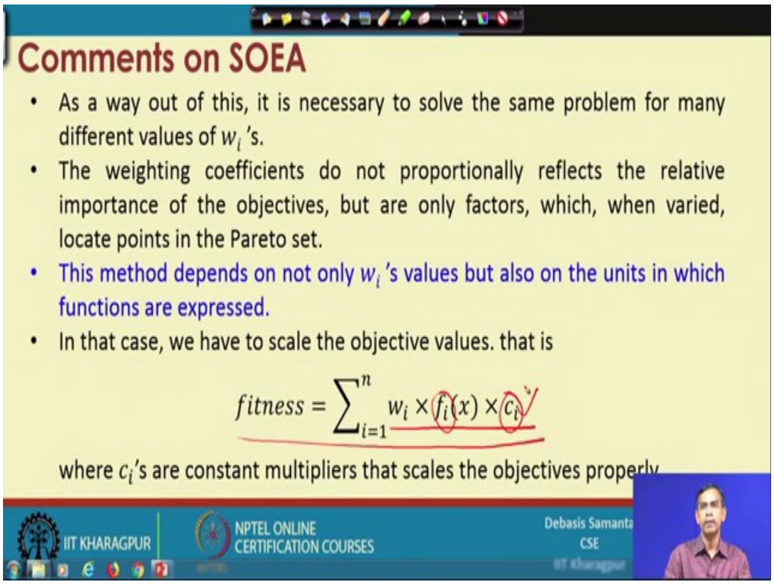
(Refer Slide Time: 28:14)



Now, so, we can say that if we do not select the w_1 and w_2 precisely then many Pareto optimal front are possible and then which solutions are the correct they are not necessary the local they are not necessary the global solution at here. So, our objective is to find the right values of w_1 and w_2. So, that if this is the solution space there it will find the solutions like this one so, that any one solution can be obtained as a solution for the multi objective optimization problem.

So, what you can say that, the usually if we don't select the coefficient values correctly then it leads to a local optimal solution. Now this is our major drawback of this technique Single Objective Evolutionary Algorithm.

(Refer Slide Time: 29:08)



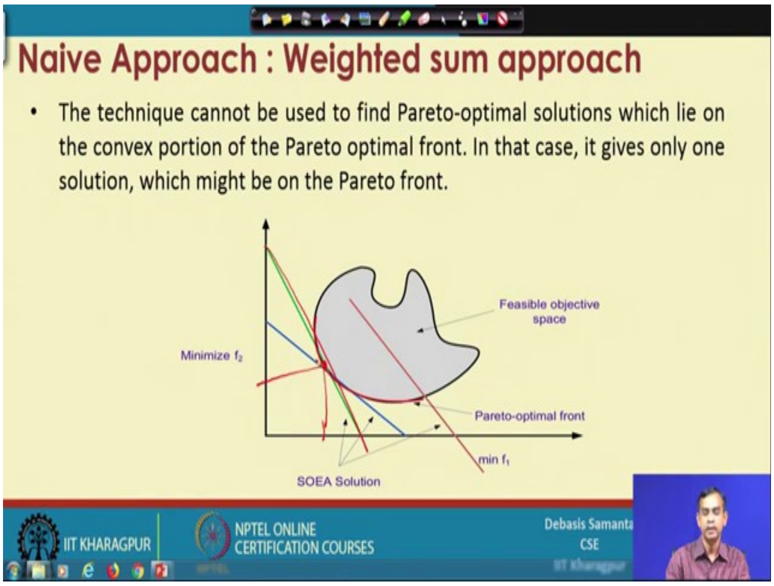
So, the idea it is like this. So, what is the possible remedy to solve this problem. It basically to solve the same problem for the different values of weighting coefficients. However, the weighting coefficients as it is infinite sets are possible. So, try out with many possibilities are also not computationally feasible.

And so, weighting coefficient do not proportionally reflect the relative importance of the objectives that is in case of lexicographic order you know, but are only a factors which one vary it can locate the point on a search space in different sets. Now the methods, depends on not only the decision of the right values of the weighting coefficients, but also importance on the units in which the different functions are expressed.

Now, so, suppose the two objective functions are there one unit is in the millimetres scale and another is in the kilometre scale then definitely weighting formula that can obtain using this formula will be the effective one. So, it required that all the objective functions are to be expressed in the same scale, as a way out these things the idea about that one scaling factors for each objective functions needs to be multiplied in addition to the weighting values weighting coefficients and for say i- th objective f_i let c_i is a scaling factor it is multiplied by this then the fitness function can gives the proper meaning or the effective meaning.

So, the c_i is called it is basically is a scaling factor in order to normalize the values of all objective function or in other words is a scaling factor. So, that all objective functions can be expressed in the same scale if it is possible. So, this is the idea about it although it is not an issue it required little bit processing. So, that we can understand the, what are the scaling coefficients for each objective function in the problem. So, this is the idea otherwise the SOEA approach is very effective and useful.

(Refer Slide Time: 31:46)



And, so these approach as I told you is a very Naive approach and then ascent approach we can and also it is termed as the weighted sum approach and we have discussed sum pros and cons about it and another advantage another limitation of this approach is that. So, if suppose the Pareto optimal front lies on a straight line it will find one solution correctly.

Suppose if the Pareto optimal front does not lie on the straight line rather in a convex or in a non-convex region whatever it is here. So, Pareto optimal front it is here or it is here or it is here, whatever it is there, then what will happen. So, in this case it requires only one it returns only one solution because if it a Pareto optimal front if it the Pareto optimal front according to the SOEA approach, but essentially the Pareto optimal front it is there it will find only one solution.

However, this solution is unlikely to get because if you find f_1 and f_2 it is here then it is basically non global solution it is not a solution actually because the solution space is this one. So, it can give some solution which is effectively not a solution rather it does not lie in a solution space. So, it can give only solutions where this line touches this Pareto front only one solution and for which you have to precisely decide what is a f_1 and f_2 , then only you able to find it.

So, it is very difficult because it is usually gives a f_1 and f_2 in this region may be which are not necessary lie or within the range all the visible solutions. So, this is why the solution that it will written the SOEA approach not necessarily be a feasible objective space rather it is a non-feasible solution it can written this is a serious drawback this is a serious drawback of the SOEA approach.

Now, so, this is the concept that we have discussed about the two a priori and then non Pareto based approach namely lexicographic ordering and then SOEA approach we have discussed. The another Non-Pareto based approach it is called the vector evaluated genetic algorithm. It is also a priori approach we will discuss in the next it is it is not an a priori it is a posterior approach rather we will discuss in the next class. So, it is a posterior approach and then Non-Pareto based approach in fact. So, we will discuss in the next class.

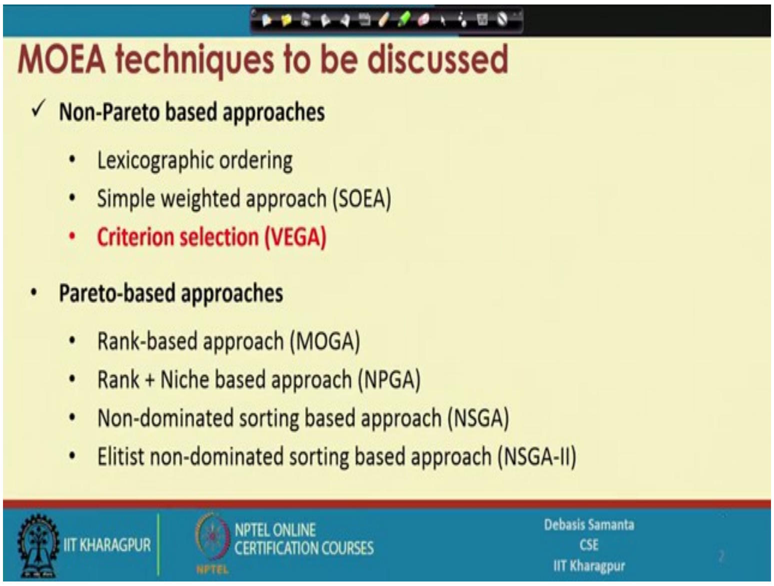
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 28
Non - Pareto based approaches to solve MOOPs (Contd.)

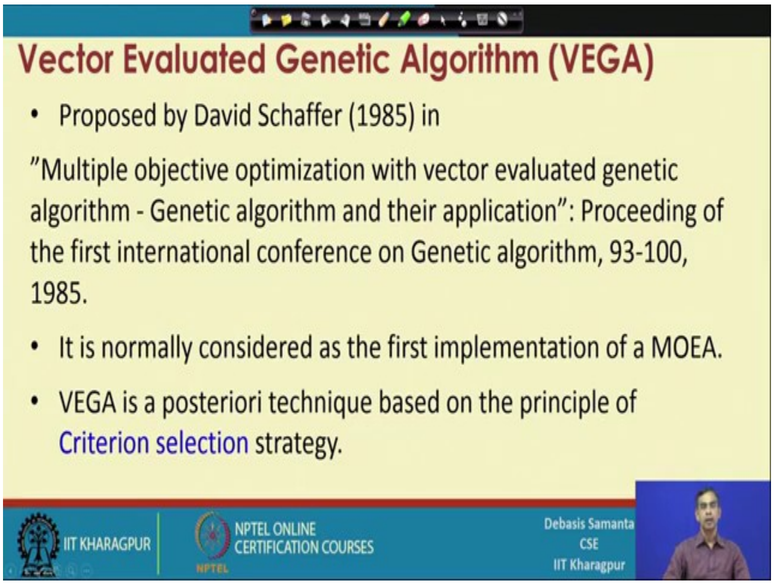
We are discussing non-pareto based approaches to solve multi objective optimization problem. In the last few lectures, we have discussed two approaches. These are non-pareto based approaches namely, lexicographic ordering and simple weighted approach.

(Refer Slide Time: 00:33)



Shortly, it is termed as SOEA. Another approach in this line is called vector evaluated genetic algorithm. Short term is called VEGA. This approach is totally different than the previous two approaches. So, this approach is basically called the based on criterion selection. So, that is why, it is a criterion selection based approach. So, in this lecture, we will learn about the VEGA approach to solve multi objective optimization problem.

(Refer Slide Time: 01:23)

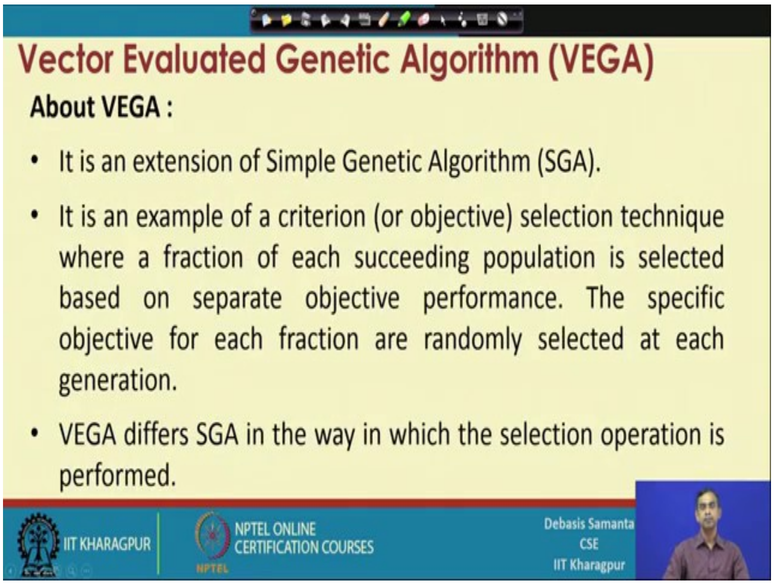


Now, this approach, first time proposed by David Schaffer in 1985. He reported the approach. The title of his approach is called multi objective optimization with vector evaluated genetic algorithm- Genetic algorithm and their application.

It is published in the proceedings of the first international conference on genetic algorithm in 1985. It is in the same proceedings where, the first approach on the lexicographic ordering was also proposed. Now, so this approach, this VEGA approach is called the fast of it is kind or it is a fast implementation of the MOEA approach.

This approach in fact, unlike the previous two approaches, it is a posteriori technique. That mean for this approach, we do not require any prior knowledge. So, this is the one difference among the three approaches which are based on non-pareto based approaches. And another characteristic of this approach is that, it is the criterion selection based approach; that means, here selection is different than the selection technique that is follows in a simple genetic algorithm.

(Refer Slide Time: 02:57)

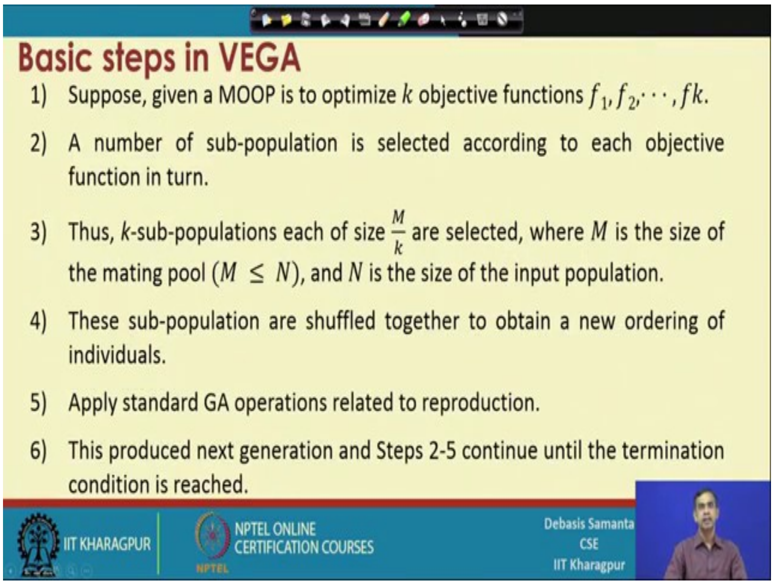


More interestingly, this approach the VEGA approach, in fact, it is an extension of simple genetic algorithm. It is an extension because, it considers a different selection strategy then the selection strategy; that is, there in simple genetic algorithm.

Now, we will discuss exactly what are the selection? What is the selection strategy that it follows in this approach? Now, here basic idea about is that, as you know, in case of multi objective optimization problem, there are multiple objectives. So, VEGA considers one objective at a time out of the several objectives in succession. That mean, it will consider one objective, then next objective and so on and then, it is the selection. Selection of a particular solution is based on the performance of the objectives as a whole. That is different; they are with respect to the simple genetic algorithm.

So, here basically, a particular objective out of the several objectives are randomly selected while, we generate the population. Anyway, let us proceed to learn about VEGA and then we will be able to understand how the selection strategy is different in this here.

(Refer Slide Time: 04:28)



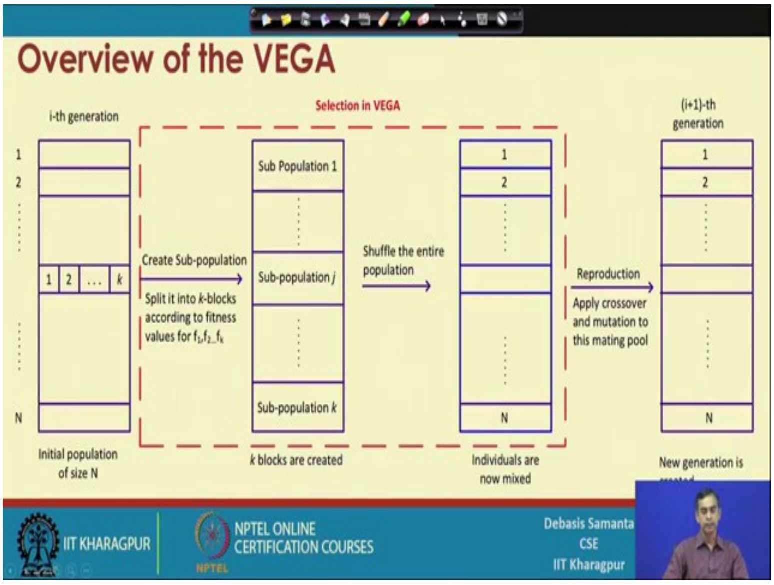
Now, we can start with let us consider, there is a multi-objective optimization problem with k objective function. So, k objective functions we denote them as $f_1, f_2, \ldots, f_k$. So, these are the objective function that we will consider here k number of objectives are there. Now, the next step is that, we have to select a number of sub population and the sub population means subset of the populations and this subset of population is selected according to each objective function in succession.

Therefore, if there are k objective functions are there. So, we have to select k sub populations. Now, size of each population is M/k, where M denotes the size of the mating pool and usually $M < N$, where N is the size of the population. So, here basically the idea is that, out of N solutions in the population, current population, we have to select M solutions for the mating pool and from the mating pool we generate the mating pair and therefore, reproduction. Now, so the second step, as I told you, we have to select k number of sub populations; each of size M/k. Once a sub populations are selected, then we have to follow one shuffling technique. So, here basically shuffling means, we have to shuffle from i-th sub population to j-th sub population, so that some mixing is possible.

Once this is done, these are mating pool is ready and then, we use this mating pool to produce the next generation. So, this is the idea about it and then we will continue the same procedure until we reach the termination condition.

So, these are the simple steps that is there in the VEGA approach. Now, let us explain the VEGA approach how it works.

(Refer Slide Time: 07:09)



Now, this figure will help us to explain the concept that just now I have mentioned here. So, the idea it is like this. So, suppose, this is the current population at any time of the generation iteration.

So, let us suppose the i-th iteration is going on. So, in the i-th iteration, the current population, this is the current population and then the size of the current population is N. And as I told you, this is an multi objective optimization problem with k number of objective vectors. So, this is the k number of objectives are there now. So, given this is the population at any instant. Our next task is to create sub population. So, create sub population like this. So, here basically, if this is the entire population, then we have to select a block of size M/k.

So, there are these way k number of blocks are there. So, each blocks are there. So, k number of blocks are there. Now, in each block, we have to select the solution which has the highest values with respect to a particular objective function. For example, in block 1, from this solution, we select all the solutions which has the highest value of objective vector f_1. So, it is basically f_1 and in this is f_j and this is f_k. This means that, in each block, all the solutions are having excellent with respect to one objective vector only.

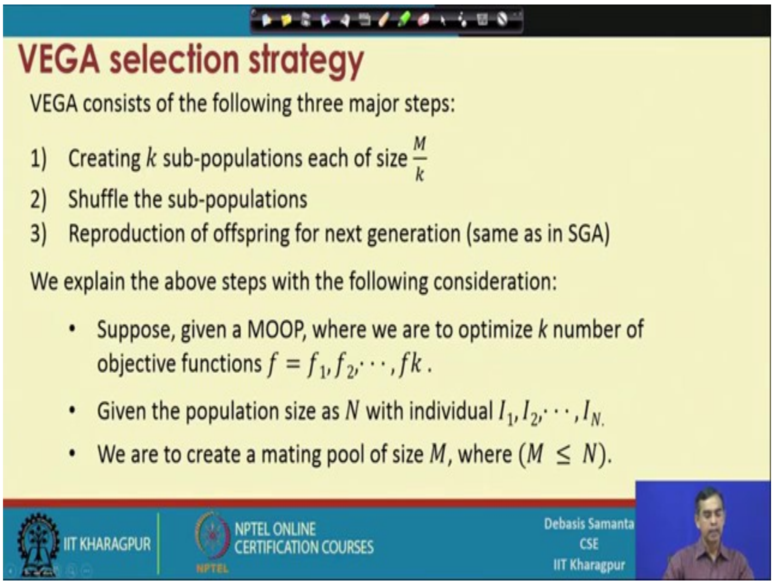
So, all the solutions here are very good with respect to objective function f_1. In the j-th block, all solutions are excellent with respect to the objective vector f_j. And in this k th block all solutions are very good with respect to the objective function f_k. So, this is the idea here. So, here M/k number of solutions, here M/k number of solutions and M/k number of solutions.

So, these way, we have to create k number of sub populations. Once the sub populations are created, then we have to shuffle. Shuffle means, one solution from here will be shuffled with to one solution in the population block j or one sub population from here to this one or any other solution from here to here. So, from any i-th block to any j-th block some shuffling was carried out. So, this is the shuffling. After shuffling, it will give a mating pool.

So, this is the mating pool, where, few are very good with respect to objective function in particular region and then other will inferior. So, it is the idea about it. And then, once this mating pool is created, we follow the same technique of the reproduction that is there in case of genetic algorithm. And from this mating pool, we will be able to create the next generation population for the next generation the $(i+1)$ -th generation.

So, this is the idea about the VEGA. And we can understand that VEGA has the different approach. So, far this technique, selection is considered. Now, we will discuss about how the sub population is created and then how shuffling occurs and then how the VEGA can complete a cycle and all this thing.

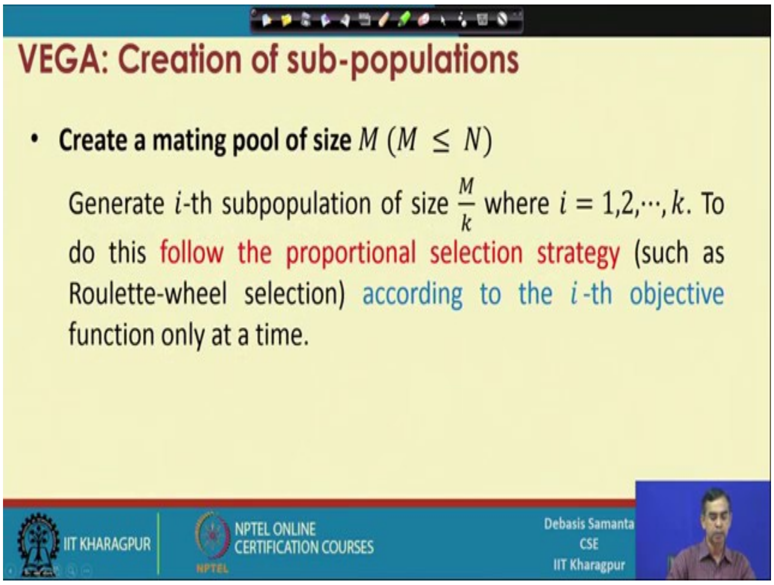
(Refer Slide Time: 11:23)



So, here VEGA consist of three major steps as we have planned. So, first step is basically creating k number of sub population each of size M/k. The second step that, shuffling the sub population and then finally, the third step is the reproduction to produce offspring in the next generation. And as we have discussed about that, the first step is different, second step is different and then this third step is the same as the simple genetic algorithm that we have planned.

So, this is the idea about it. Now, let see how the different things or shuffling or sub population can be created and then, what is the procedures there.

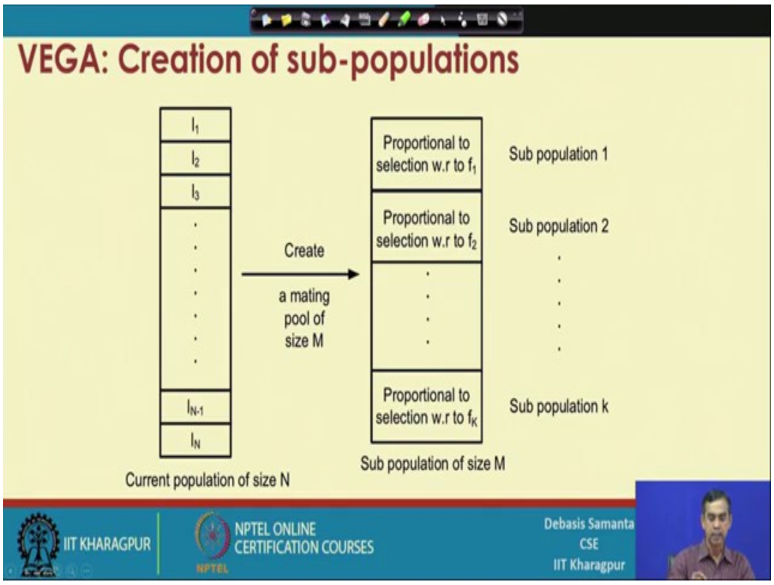
(Refer Slide Time: 12:10)



Anyway, so, once the mating pool of size M is created, we have to follow the proportional selection strategy. So, this is basically, the idea about how to create the sub population. So, to create the i-th population of size M/k from the entire population, where i is basically $1,2,\ldots k$ any one numbers, here to select the sub population for the i-th block, we follow any selection strategy.

For example, any proportional based selection strategy such as, roulette-wheel or rank based selection can be followed. So, here, whenever we follow this roulette based selection strategy, like this one, basically we will consider only one objective vector, one objective function; that means, so i-th objective function should be considered in order to select the proportional selection strategy to fill the i-th block or i-th sub population. So, this is idea about how to create the sub population any one population like this one.

(Refer Slide Time: 13:38)

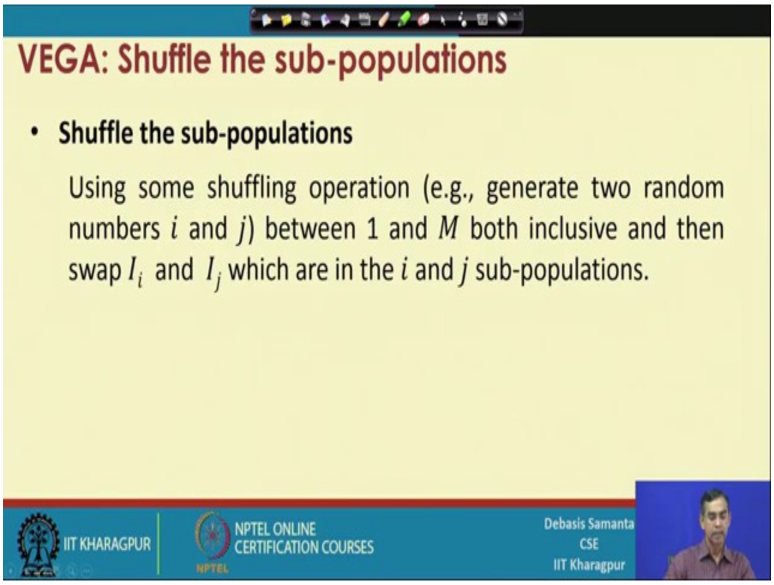


So, this is the pictorial description of the same thing. Here, this is the entire population of size N.

Then, our objective is to create the mating pool of size M. So, what we should do is that, for this for this population, when we have to fill it up, we will follow any proportional based selection strategy to select k. Select M/k number of solutions from here to obtain the sub population 1. Now, here and then, whenever we apply this selection; that means, to select all the solutions for this block, then we can follow any proportional based selection and then with respect to objective function, say f_1. So, here with respect to objective function f_1, we apply any proportional based selection and then sub population will be created.

So, this will be continued with respect to f_2 for the sub population 2. Similarly, with respect to f_k for the sub population k. So, this way, we will be able to create the sub population of size M where, the number of solutions belongs to a particular block has the excellent, with respect to one particular objective function f_1 and so on. So, this is the main the main task the major one task that is there in VEGA.

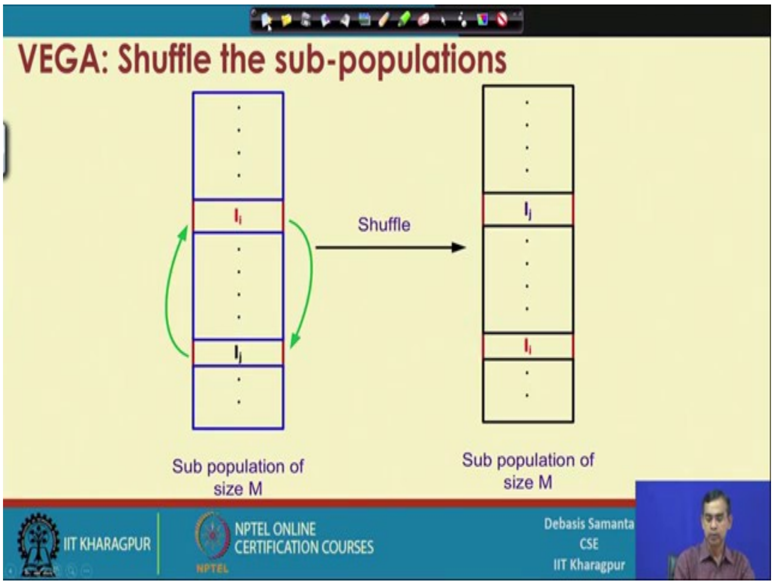
(Refer Slide Time: 15:20)



And once this sub population is created, then our next task is to shuffle the sub population. So, the shuffling that can be, we can like shuffling means, shuffling the solution from one block between any two blocks. Now, this shuffling can be done in many ways. I have mentioned one simple approaches here. For example, we first generate any two random numbers, say $i \wedge j \, between \, 1 \wedge M$.

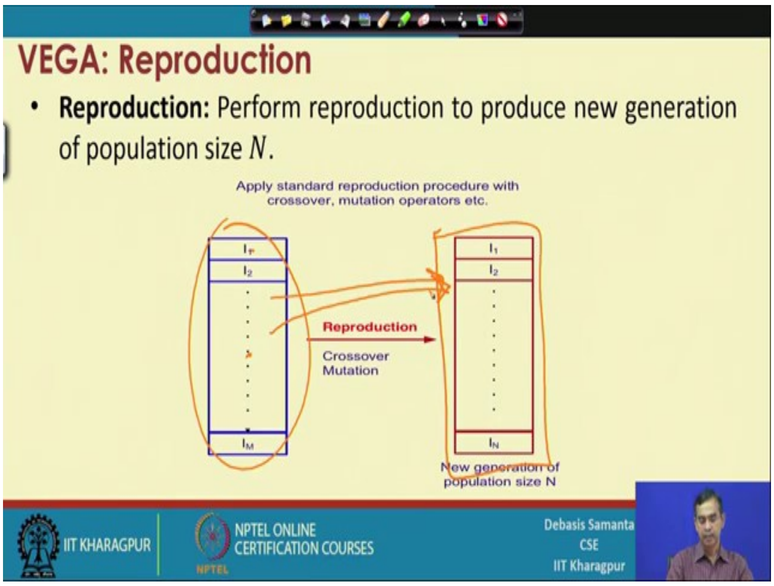
Then, swap I_i; that means, the solution I_i and then solution I_j, which are there in i-th and j-th sub population namely. So, we have to select any two solutions at random which belongs to any two sub population and then shuffle them. And this shuffling can be repeated may be some p times, where p is decided by the programmer. p may be 10, p may be 5, that means, how much mixing of the solutions you want to have. So, this is the idea about shuffling. Once the shuffling is done, then our mating pool is created.

(Refer Slide Time: 16:35)



So, here basically the illustration of the shuffling procedure, this is the one solution in any i-th block. This is another solution in the j-th block. Basically, shuffling that mean I_1 will go to the j-th population and I_j will go the i-th population. So, after shuffling this is the mating pool is there. Now, once this mating pool is known to us, we will be able to go for reproduction. Reproduction strategy is same as the reproduction strategy that is there in case of simple genetic algorithm.

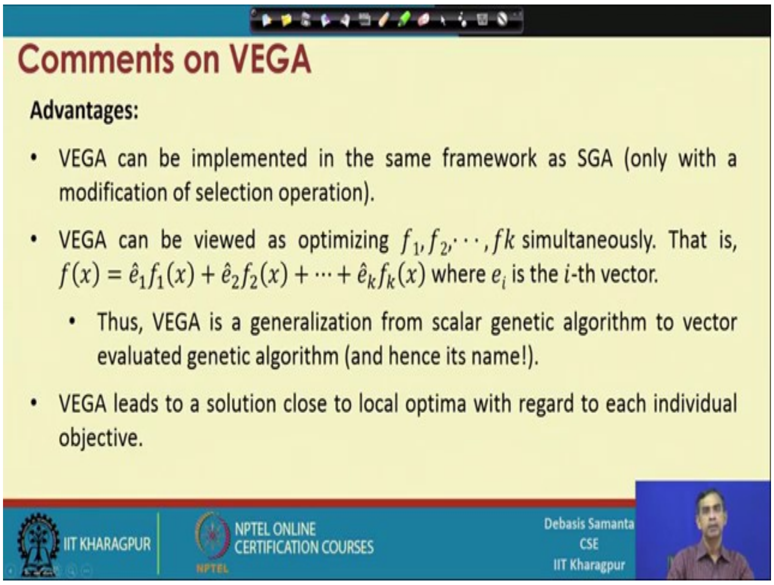
(Refer Slide Time: 17:03)



Here, this is the mating pool and from this mating pool, we have to create next generation populations. For the next generation, size will be same. That is the N is the size of the population.

So, here, we can do again any two solutions at random. So, this one and this one any two solutions at random at the mating pair and from this mating pair, we will be able to generate offspring depending on the crossover operation; mutation operation that we have already learnt for the simple genetic algorithm. So, this way, from the mating pool of size M, we will be able to create the population of size N. And so, this is the idea about reproduction that is there in case of vector evaluated genetic algorithm.

(Refer Slide Time: 17:59).



And then, let see, how this vector evaluated genetic algorithm. We understand that, VEGA can be implemented in the same framework as SGA. That means, it is basically an SGA, but only the thing is, that the selection strategy that is there, we have to little bit modifies.

So, it is basically modified version or extended version of simple genetic algorithm. Now, VEGA can be viewed as optimizing all the objective function in succession and that is why, we can write that, it basically optimizing a function x, which is in the vector form of all the objective functions like $f_1(x), f_2(x)$ this one. So, this is why, this method is called the vector objective. So, it is basically, we have to find $f(x)$ such that $f(x)$ is has one component $f_1(x)$ another component $f_2(x)$ and then k

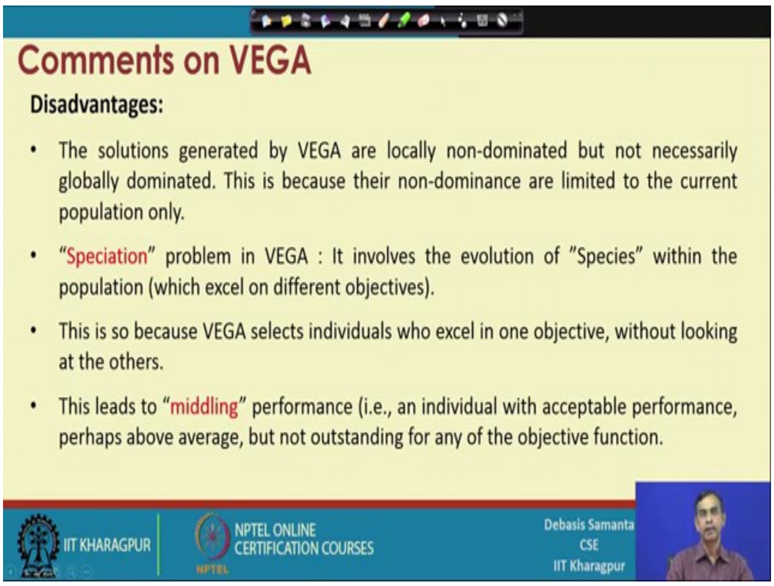
component $f_k(x)$. So, we have to find $f(x)$. So, that $f_1(x)$ is good, $f_2(x)$ is good and then, $f_k(x)$ is good. All objective functions are good.

But, we consider all objective function, whether good, then it is not at the same line rather in a succession line. So, first we say that, whether it is good with $f_1(x)$ or it is good with $f_2(x)$ or it is good with $f_k(x)$. Then we can say that $f(x)$ is good with respect to k objective functions like this one. So, this is the concept that is there and that is why, it is the vector evaluated.

It is called the vector evaluated genetic algorithm. It is in fact, a generation from the scalar genetic algorithm; scalar means, if we multiplied this f_1 by w_1, this by w_2, this by w_k and if we consider $w_1 = w_2 = w_k$, then it is basically a scalar generation. But, it is not scalar generation. It is basically vector generation and then the each component of the vector is basically excellent with respect to that component only and a thorough research and experiment with different case studies, it is observed that, this VEGA approach is comparable to the previous two approach; namely, lexicographic ordering and then SOEA approach.

However, and then SOEA approach, it is better compared to the previous two approaches and it also leads to a solution very close to local optima with regard to each individual objective vectors are there. So, this approach can be considered as a better approach.

(Refer Slide Time: 21:14)



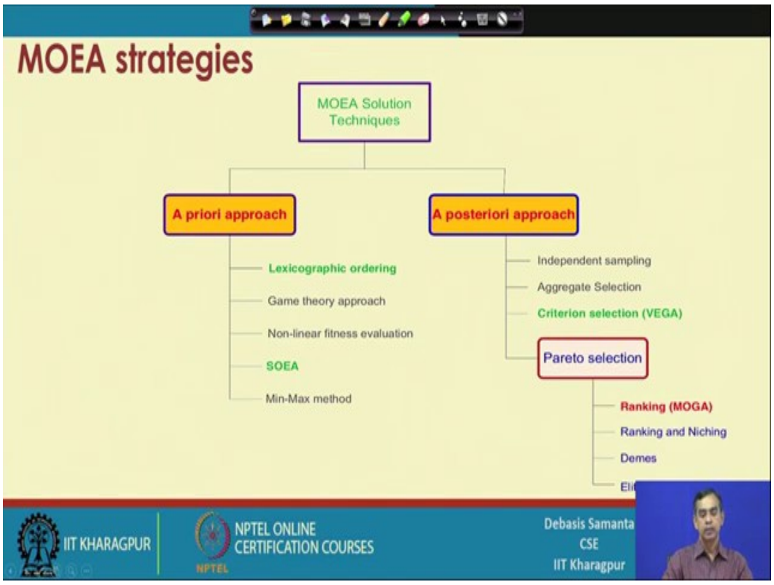
So far, the approaches that we have learnt and so, this is the advantage that, the simple genetic algorithm framework can be easily adapted to implement the VEGA approach and it gives better result compared to the previous two approaches. However, it has certain serious limitations and fine like the previous two approaches, which are belongs to non-pareto based approaches, this approaches also gives only one solution.

That is why, it is called a non-pareto based approach. Now, the solutions generated by VEGA are locally non-dominated, but not necessarily globally dominated; that means, all solutions are close to the local solution, but not necessarily close to the global solution. So, it cannot return the global solution. In fact, another two problems with this approach; the problem here called as speciation problem and middling problem.

So, this are the two drawbacks in this VEGA approach. Now, we say that, it suffers from the speciation problem, if it basically considers or if basically do very good result in terms of a particular objective vectors, but not all objective vectors. This is why, and now, in case of VEGA approach, we see that, whenever we select a solution for the next generation, we basically select with respect to a particular objective vector. Now, so that is why, it suffers from the speciation problem. That means, whenever we select an object we don’t consider all objective function at a together, rather all objective function at a time. So, this VEGA approach suffers from this speciation problem.

Now, another problem that the VEGA approach suffers is called the middling performance. Middling performance means, the results that it returns neither very bad nor very good it is a say middle performance; that is, the things is there. So, if we are interested with not so accurate result. So, for the objective function optimization is concerned, we can use VEGA approach. In fact, VEGA approach is very fast compared to any other multi objective optimization solving approach. So, this is the pros and cons of the VEGA approach that we have learnt.

(Refer Slide Time: 24:02)



So, we have learnt different techniques to solve multi objective optimization problem. We term all the approaches as the MOEA algorithms or MOEA strategies. And so, far the MOEA strategies are there, we consider non-pareto based approach. So, all the solutions are non-pareto based approach. And out of these non-pareto based approach solution, the solution that we have discussed as A-priori approach. So, we have these are the A-priori based approach.

And so, and then, in this course, we have learnt about two A-priori based approach; namely, SOEA and the lexicographic ordering. They are non-pareto and A-priori approach. Another non-pareto based approach that we have just now learned it the VEGA approach. The VEGA approach, in fact is A-posteriori approach. So, there are different variations. So, far the non-pareto based approach are concerned, we have learned it and then we will we are going to learn about the pareto based approach.

There are many good techniques or good approaches belongs to this group pareto based selection. So, we will discuss about one approach all. So, we will discuss about four approaches in this course ranking and then niching the demes and then elites. So, those things will be discussed in the next slides, next class.

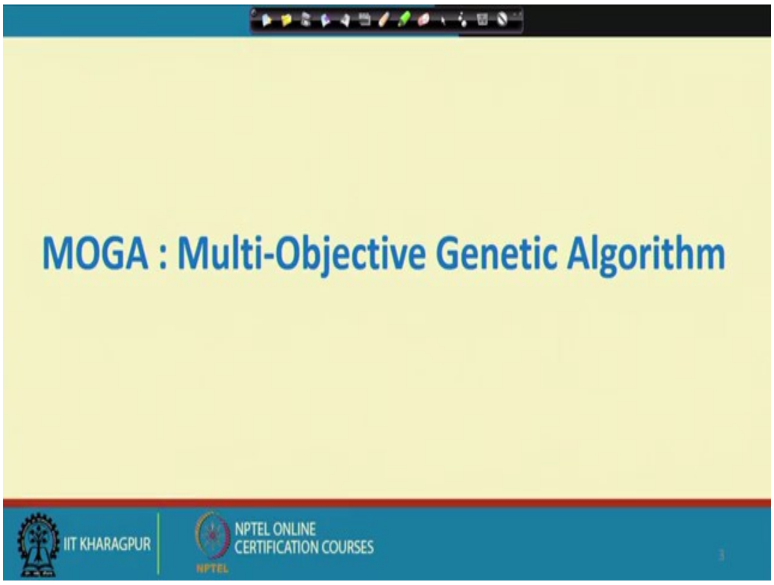
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 29
Pareto-based approaches to solve MOOPs

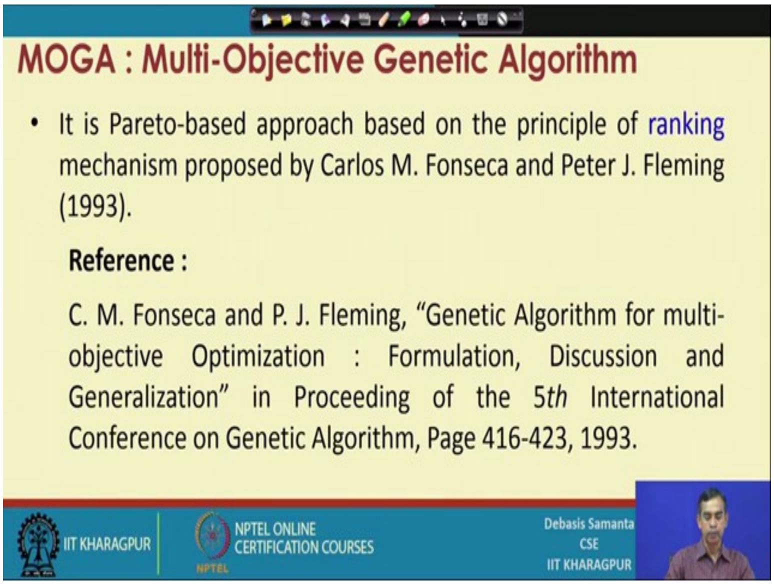
In the last lecture, we have learnt one Pareto based approach which is also a posterior approach called the VEGA. In this lecture, we will learn about other Pareto based approaches. The first we will discuss about the MOGA. The MOGA short form it is called multi objective genetic algorithm. It is a Pareto based approach and also it is a a-posteri based approach, because no prior knowledge is required to solve this problem.

(Refer Slide Time: 00:40)



(Refer Slide Time: 00:59)

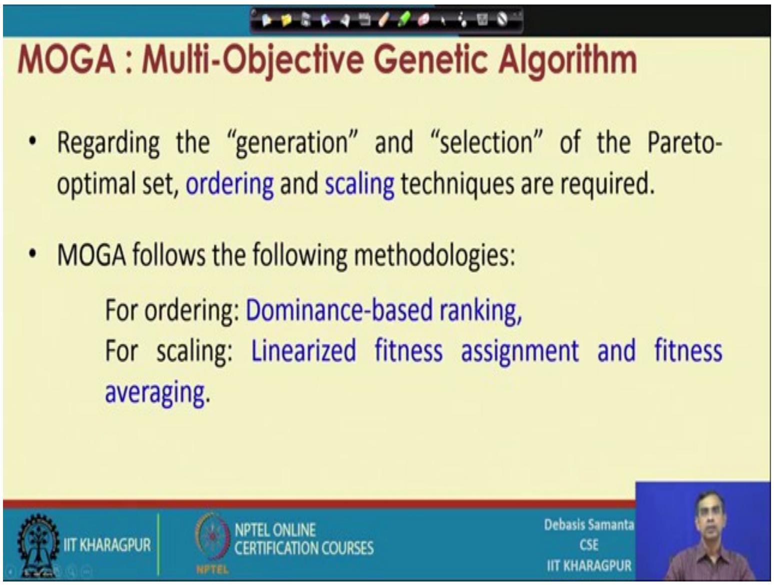
J



Now, this approach this Pareto based approach first time proposed by Fonseca and Fleming in 1993, they published one work the title of the work was genetic algorithm for multi objective optimization, formulation, discussion and generalization published in the proceedings of 5th international conference on genetic algorithm.

This conference was treated as a best conference in the field of genetic algorithm. Now, Fonseca and Fleming proposed this approach and the basic principles behind this approach is ranking mechanism. So, we will learn exactly what is the ranking mechanism? And what are the different steps, that is there in this approach?

(Refer Slide Time: 01:51)



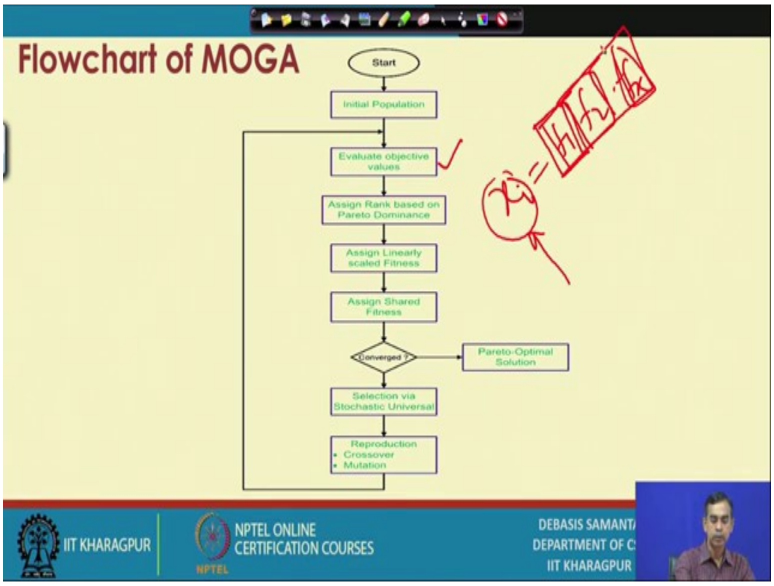
J

So, here basically the idea about I told you that, all the approaches are same as the genetic algorithm except the selection mechanism. Now, the selection mechanism towards the next generation population generation or it is basically and it is the generates the next population, so that all the solutions are the non-dominating solution; that means, they are the best solutions.

Now, here regarding the generation and selection of the Pareto optimal set; that means, non-dominating set the approach that is the MOGA approach considers two techniques the techniques are called ordering and scaling. So, here for ordering they follow one new concept called the Dominant-based ranking, and for scaling they proposed one idea it is called the fitness assignments and that is also sometimes called the fitness averaging is a linear function of the fitness averaging.

So, learning of MOGA is basically to understand clearly how the ranking is carried out there, and then how they do the fitness scaling or scaling of the objective functions. So, we will learn these two steps in the next few slides, and then the MOGA approach can be understood clearly.

(Refer Slide Time: 03:23)

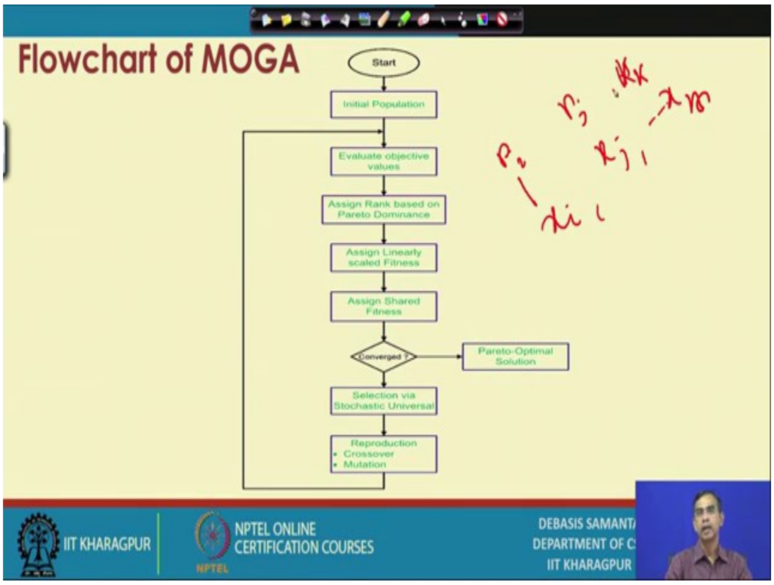


Now, here first see what is the flowchart of the MOGA approach. So, it is basically same as the genetic algorithm flowchart if we see. So, it starts with creating the initial population can be created once the chromosome is decided and it can be created with some random solutions into it.

So, the initial population creation, once the initial population is created it will evaluate the values or object or evaluate each solution. So, it is basically evaluation step which basically evaluates all the solutions, now again when you evaluate this one it basically evaluates all objective values; that means, if a solution x_i is there. So, it basically evaluates $f_1, f_2, \ldots, f_k$, if the k vectors are there.

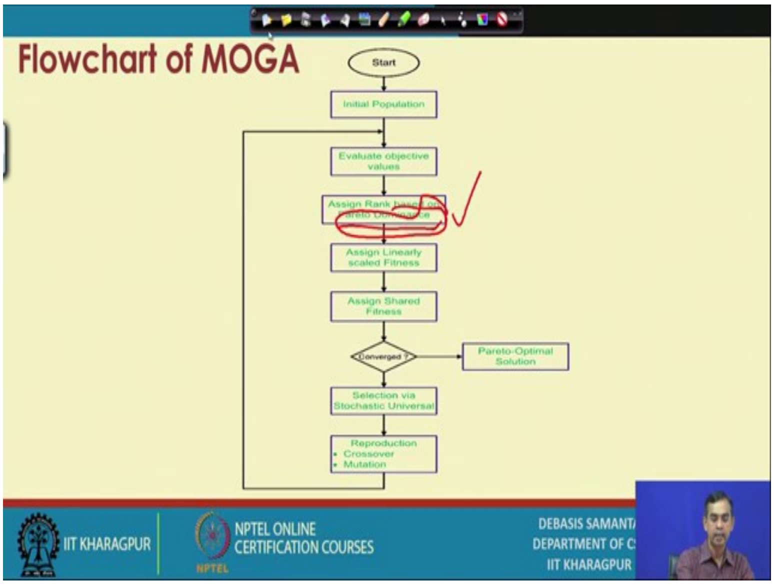
So, from the given solution x_i , where the chromosome is known to us, then we will be able to create or evaluate each objective function values. So, this is the creation and once the objective values are created then each objective value. So, x_i , x_j …. x_m , there are m number of solutions are there, then they assign rank to all the solutions. So, the rank means; it can be assigned rank i , it can be assigned rank j , it can be assigned rank k like this on. So, rank will be assigned to each solution, there may be two or more solutions can be assigned same rank and so on, but no two solutions will be assigned two different ranks. So, this is the concept that is followed here.

(Refer Slide Time: 04:48)



(Refer Slide Time: 05:24)

J



It is a rank basically assigning the rank and this assigning the rank as it is told here, based on Pareto dominance. So, we will learn about what is Pareto dominance? And this concept can be applied to assign the rank.

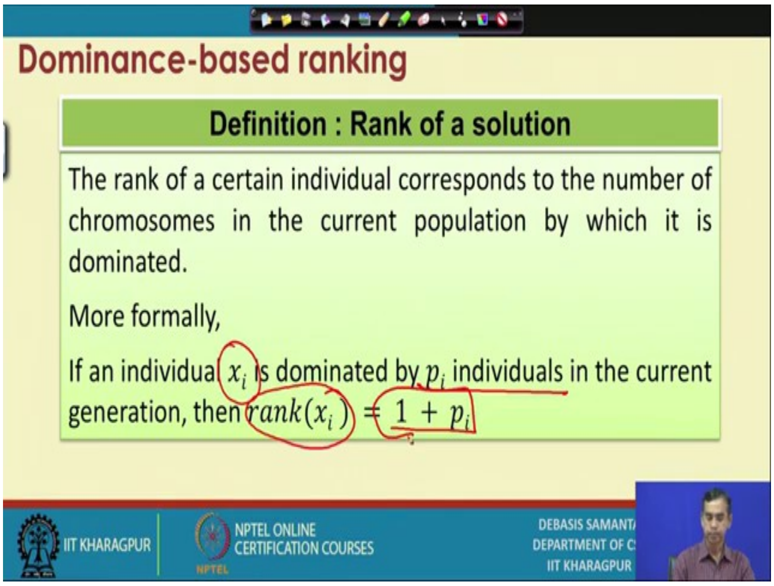
Once the rank is assigned in the step our next task is to assign the scaling. So, this scaling assignment is basically follows a linearization of all the objective vector. That means, it will follow certain linear function. So, that all solutions belong to a particular rank can be assigned one unique fitness value. So, this is the concept that is here to, and then it is called the assigning linearized scale scaling fitness, and then assign shared fitness value. So, it is basically after the linearization, we basically give the same fitness values to all the solutions, which belongs to a particular rank. So, that is why it is called the shared fitnessing shared fitness concept.

So, this will give all the solutions, but a modified fitness values like this one. Then the solution that we have to undergo certain convergence test and if they pass the convergence test, then all the solutions that we will be obtained are written as a Pareto optimal solution and if it is failed the convergence test is not successful, then we will go for the selection by means of a some probabilistic selection, whatever population based or proportional based selection, whatever the selection that we have learned about it, but it is basically a stochastic selection we will follow, and then this stochastic selection will produce a mating pool, and then from this mating pool we perform the reproduction operation and then next generation will be produced.

J

So, this will be repeated again and the cycle will be continuing to till the convergence test is convergence criteria is met. So, this is the idea about the MOGA approach and we can learn, we can understand that we can see that this MOGA approach has the basic framework same as the genetic algorithm framework, but there are few steps that is unique here, it is here, so basically assigning rank and then linearization. So, these two tasks are different for the selection is concerned or prior to the selection of course. So, it basically makes ready that, how the selection can be carried out properly, so that the non-dominating solutions can be selected for the solution.

(Refer Slide Time: 08:25)



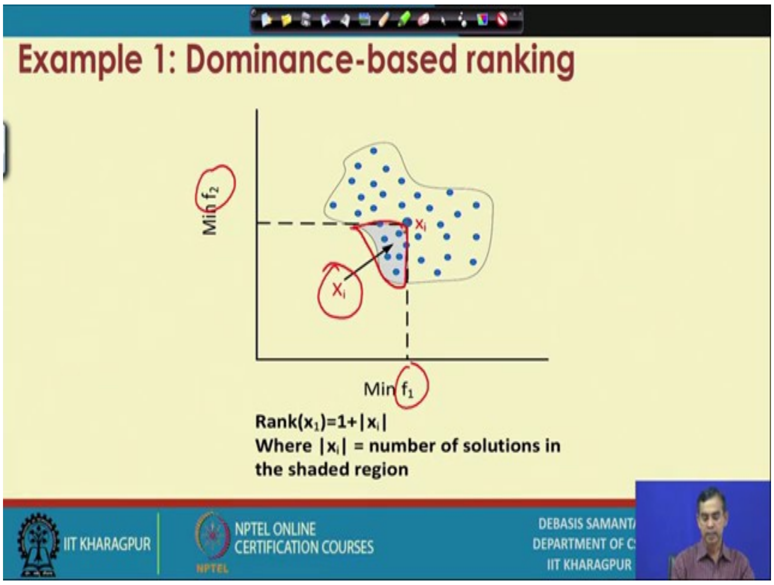
So, this is the flowchart that is followed in MOGA approach. Now, we will discuss about how to assign rank to a solution. So, rank of a solution, now here in this MOGA approach, they proposed one criteria or a one concept like the they told that, they assigned a rank like this, they the rank of a certain individual corresponding to the number of chromosomes in the current population by which it is dominated.

So, this is the concept that means, if a solution is dominated by say n number of solutions, then we can assign the rank accordingly; that means, is proportional to n like. If a solution is not dominated by any other solution, then its rank will be the lowest one. So, according to this idea about it they defined the ranking of a solution like this, if a solution x_i is dominated by p_i number of individuals in the current generation, then the rank of x_i is $1+p_i$.

J

So, this way we can easily understand that the rank of the solution which is not dominated by any one is the lowest and the lowest rank is 1. So, this is the formal specification by which rank of a solution can be assigned. Now, let us illustrate the concept with some examples.

(Refer Slide Time: 10:17)

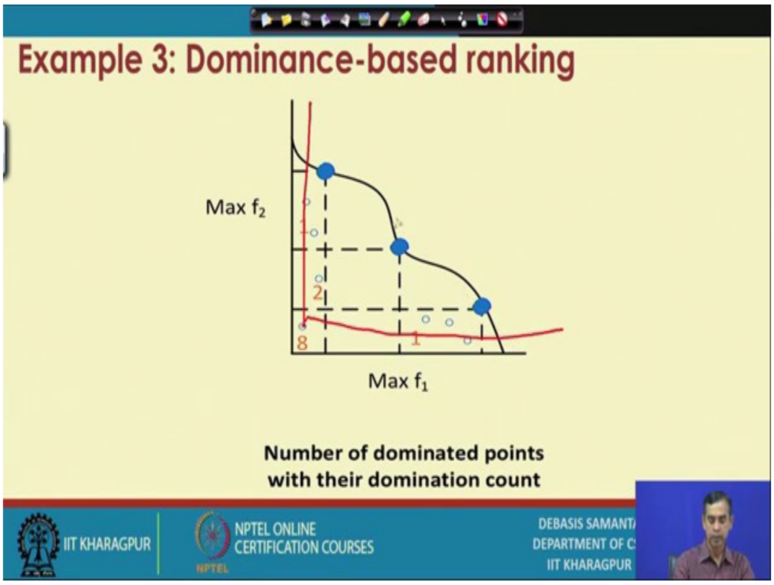


Now, say suppose this is the at any instant the solution space and we want to assign rank of any solution, let it be x_i , now if this is the solution and this is the two objective optimization problem and f_1 and f_2 are both to be minimized, then with respect to x_i these are the subset of the solution, which we can obtain solution, which we can denote it as a capital x_i and we see that these are the subset of solutions is in fact, dominates x_i or we can say x_i is dominated by all the solution.

Now, how many solutions by which the x_i is dominated, let it be x_i is a size then rank of the solution x_i it is denoted as $Rank[x_i]=1+the\,number\,of\,solution$ $which\,is\,there\in this\,region$. Now we can note that all the solution, whatever the, this region are basically dominates the solution x_i . So, this is the concept by which the ranking can be assigned and we can do it writing a very simple program. So, it is not an issue. So, the idea it is like this now. So, this is the concept if it is minimizing f_1 or minimizing f_2 .

(Refer Slide Time: 11:52)

J



Now, visually the same thing can be explained, if it is maximizing also. So, in that case, if this is the solution x_i for which the rank has to be determined, then all the objective all the solution which is in this region are basically dominates this x_i or x_i is dominated by all the solution which are here. So, the rank will be accordingly the number of all the solutions in this region + 1 that is the rank of x_i. So, this is the concept it is there.

Now, in this particular example, as you can say this solution is dominated by 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11. So, the rank is, the rank of this solution x_i is 12. So, this way it is the idea it is there. So, the rank of the solutions can be obtained and can be assigned. So, this way ranks of each and all the solutions can be assigned.

Now, for an example this solution now, this solution is not dominated by any other solution. So, rank of this solution is 1. In other words, rank of all the solution which is lying on this font is rank 1. So, this solution we can say this is also nothing, but the font is called the Pareto optimal font. So, we can say that, all the solutions which are having the lowest rank they are the Pareto optimal solution or they lie on the Pareto optimal font. So, this rank assignment can help us to know which solutions are lying on the Pareto optimal font like this one. So, this is the concept of ranking.

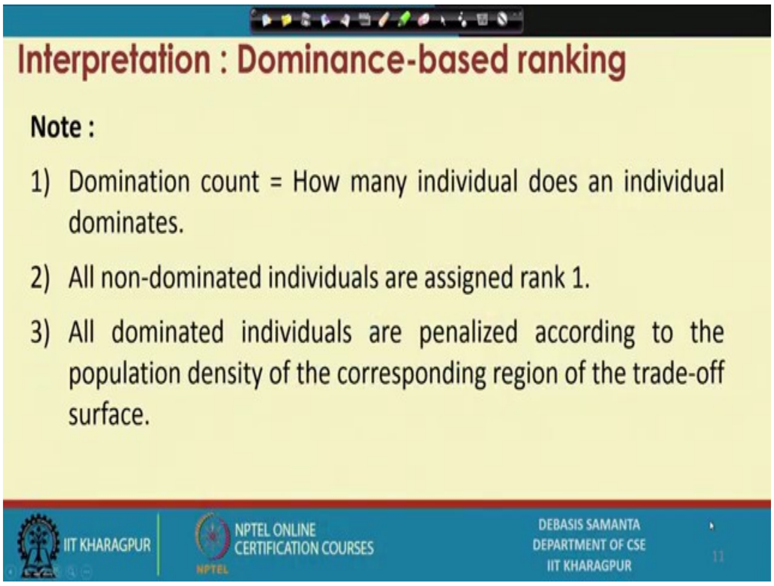
Now, another example that I can post it here, we can easily understand that it is both max f_1 and this one and if we say this solution we have discussed about. So, here basically all the solutions this is the one solution, like all the solution and you can easily

J

understand that, how the rank of these solution for example, rank of these all the solutions are 1, because this solution and this solution only dominated by this one and so on.

So, the rank of all the solutions can be obtained like this. So, rank of this solution it is like this and this one so, the 2 and so on. So, rank of this solution also it can be calculated counting all the numbers here, and then this one this one. So, rank can be calculated by this simple method, and so the rank, this is the idea about ordering all the solution based on ranking.

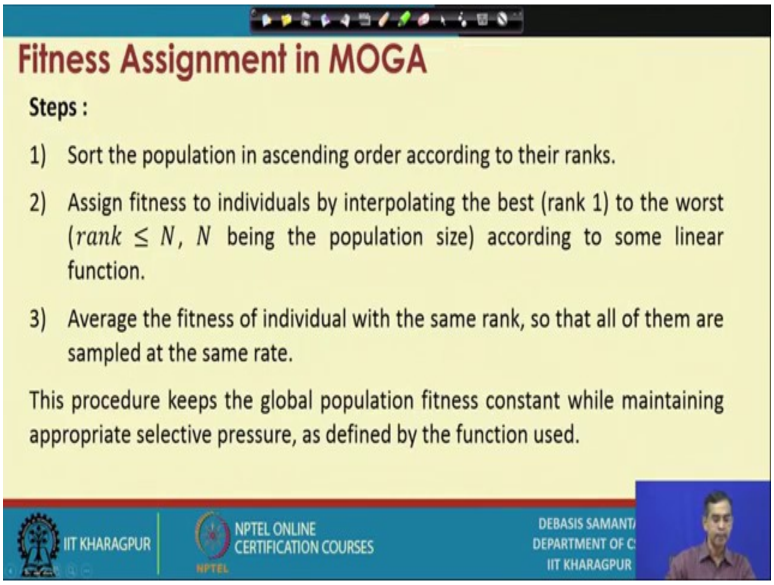
(Refer Slide Time: 14:40)



Now, once the ranking is known to us our next task is basically to linearize fitness. Now, again here ranking has certain physical interpretation, I just before going to have the next discussion, let us first discuss what is the interpretation? So, rank is basically domination count, that mean how many individuals does an individual dominates. So, this basically the interpretation, that rank can and as I told you that all non-dominated solutions are assigned rank 1 and the rank is higher; that means, they are inferior solution and that is basically a rank can be considered as a penalty, by which this solution is dominated in the population one solution having higher rank compared to the another solution; that means, it is dominated by more solution then the others.

(Refer Slide Time: 15:36)

J

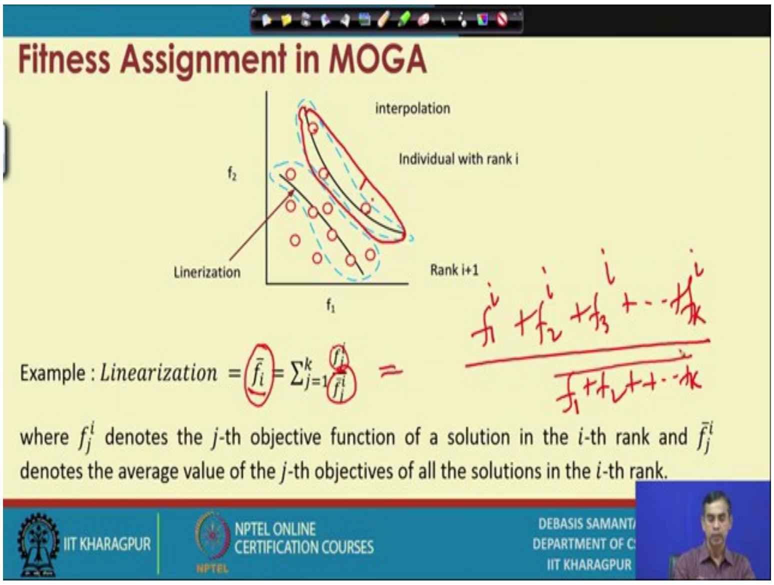


So, this is the interpretation of the concept of ranking here, and then once the ranking is done our next task in the MOGA is basically fitness assignments. Now for the fitness assignment the idea it is like this, it follows few steps three steps as we mentioned here, the first step is to sort all the solutions in a current population according to their ascending order of the ranks. So, basically, we sort all solutions based on their rank actually in ascending order.

Then we assign the fitness to individual by interpolating the best rank to the worst rank line. So, worst rank may be as close as N, if N is the number of population size, then we assign all solutions belong to a particular rank in terms of some linear function, we will discuss what exactly the linear function that it follows to linearize the solutions fitness values. It is basically is a linearization followed by averaging the fitness value. So, this is the main idea about fitness assignment to each solution.

(Refer Slide Time: 17:03)

J



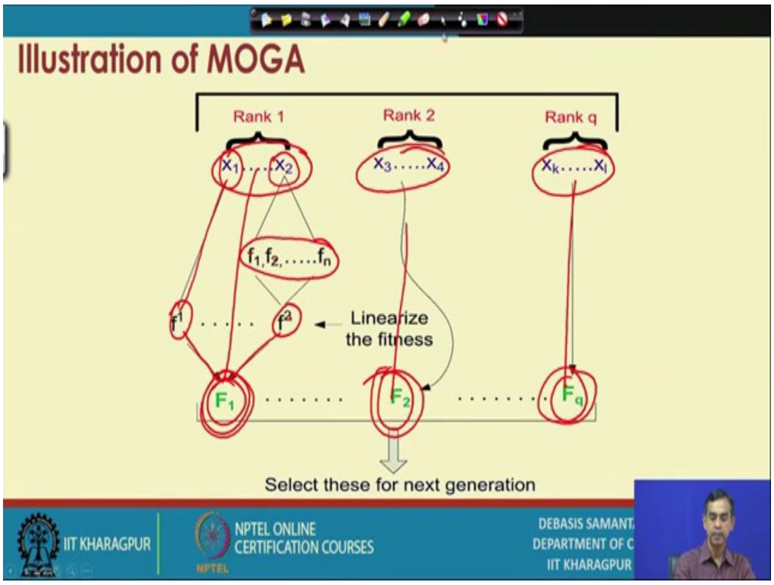
Now, let us understand this concept with an example better we can follow some example and then we follow it. So, idea about fitness assignment I am discussing. So, idea is that the basic concept behind this or rational behind this approach is that, say suppose all these solutions are assigned one rank, then basically the idea is that all the solution which are assigned rank 1 should have only one fitness value like; that means, all solutions belongs to a particular rank has the same fitness values. So, to do these things it first expresses all the objective into some linear function here the idea it is like this. So, linearization is like this it is basically, linearizing all the solutions belongs to the i-th rank.

We denoted that $\acute{f}_i$, then for all the solutions which is there suppose it is the k number of objective functions are there, then for with respect to each objective function we take the sum of all the objective functions divided by the average objective functions. So, this way we can express a linear function like this one. So, this can be like this. So, $f_1^i+f_2^i+f_3^i+\ldots+f_k^i$ and divided by. So, it is basically. So, here and f_j^i denotes the average value of the j-th objectives of all the solution. So, it is basically. So, the j-th objective if you say the j , j means $f_1+f_2+\ldots+f_k$ and then this is the average value.

So, this way it basically calculates the linearization of this objective functions. So, once the linearization is done, then the next step is basically to assign the fitness value. So, the assign assigning the fitness value will be like this.

J

(Refer Slide Time: 19:20)



Here this slides can help us to understand how it takes place? Now, first consider these are the solution, which is having rank 1 and this another solution another rank tone. So, there are group of solutions belongs to the different rank like.

So, all the solutions belong to one rank, all the solution belongs to another rank, all the solution belongs to another rank. And here the solution any solution belongs to this rank has the fitness value $f_1 + f_2 + \ldots + f_n$ the fitness functions and we can express all the solutions that belongs to this one by means a linearized one. So, f_2 similarly for x_1 this is the f_1.

So, it is called the linearization using the previous step, that we have discussed. Once the linearization is done, then we can take the average values of all those. So, it is basically averaging. So, this way all solutions which belongs to a particular rank has a fitness value f_1. Similarly, all solutions which belongs to this rank has the fitness value f_2 and all solution belong to this rank has the fitness value f_k. So, what you can understand is that, all the solutions belong to a particular rank has the one fitness value and another fitness value and this is the another fitness value.

So, this way we assign the fitness values to the same fitness values to all solutions belongs to a particular rank. That means, if you have if the solution has the same rank,

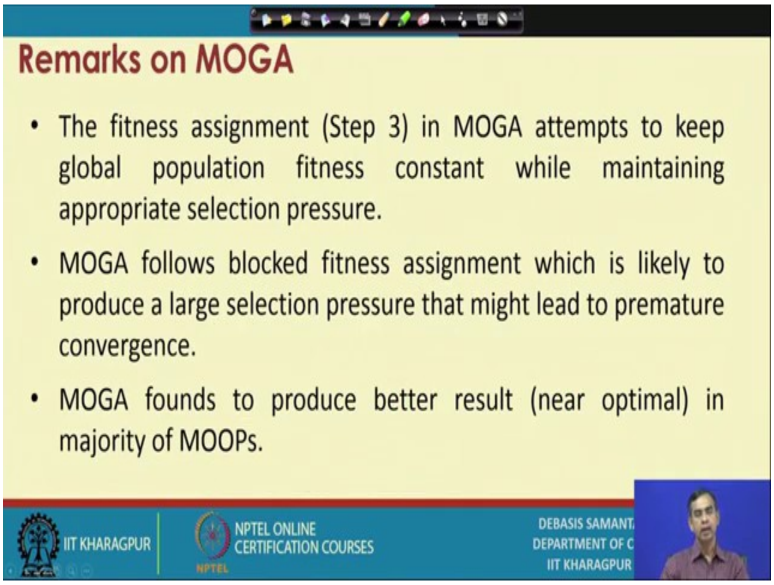
J

then they have the same fitness values. So, it is basically ranking followed by the assigning the fitness value is the step.

Now, once this fitness values are there, then based on this fitness values, we will go for selection. That means, that selection can be any selection may be say proportionate based selection, like say roulette wheel selection or rank selection or in this case one particular selection is called the stochastic selection. Stochastic selection just like a proportionate based selection, but it is basically random selection. It basically generates a random number and then in general in the range of the fitness values of their and then, it basically selects a particular solution based on the random number that is generated. So, is a stochastic this one, but other than the stochastic also, we can follow any standard selection that is used in case of simple genetic algorithm also there.

So, this selection will create a mating pool by from where the conventional reproduction method can be applied, and then next generation can be obtained. So, this is the idea about this MOGA approach here.

(Refer Slide Time: 22:32)



So, this way it can help to I mean have the dominant solution; that means, it always gives more preference to the solutions, which are basically non-dominating actually or dominated by lesser number of solutions. So, this way it searches to that direction, which basically towards the Pareto optimal solutions. Now, here the main idea about fitness assignment that we have this learned about that it basically the objective of the fitness

J

assignment is to keep the global population fitness constant, while maintaining appropriate selection pressure. That means, we will select all the solutions which has the lowest rank.

So, far the dominant a dominant solution is concerned and it also follows the blocked fitness assignment, because it is called the block because a solution belongs to a particular fitness has the same what is called the fitness values. So, this basically produces a large selection pressure and that may be sometimes to lead to premature convergence. That means, it can terminate giving non-optimal solution or a local solution.

However, it is observed that this MOGA approach founds to produce better result, near optimum or the global optimum solution in many of the multi objective optimization problem. So, this is the one approach, the MOGA approach, MOGA is basically out of the different Pareto based approach, except the VEGA approach is one of the simplest yet more effective approach known so far. Now there are many other approaches also known which are basically more elegant, more efficient and gives better result and all this approaches we will discuss in next class.

Thank you.

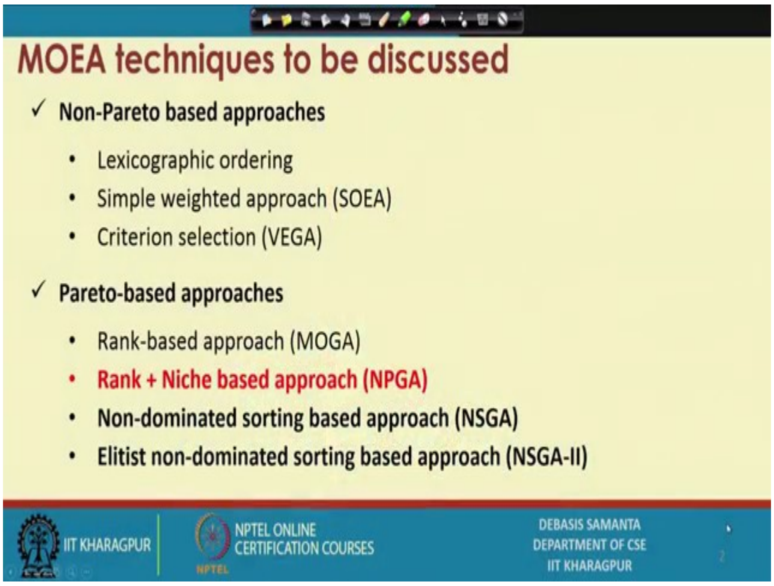
J

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 30
Pareto- based approaches to solve MOOPs (contd.)

We are discussing Pareto based approaches to solve multi objective optimization problem. In the last class, we have discussed one such approach it is called the MOGA. In this lecture, we will learn another approach it is called NPGA. It is a short form. It is full form is Niched Pareto Genetic Algorithm.

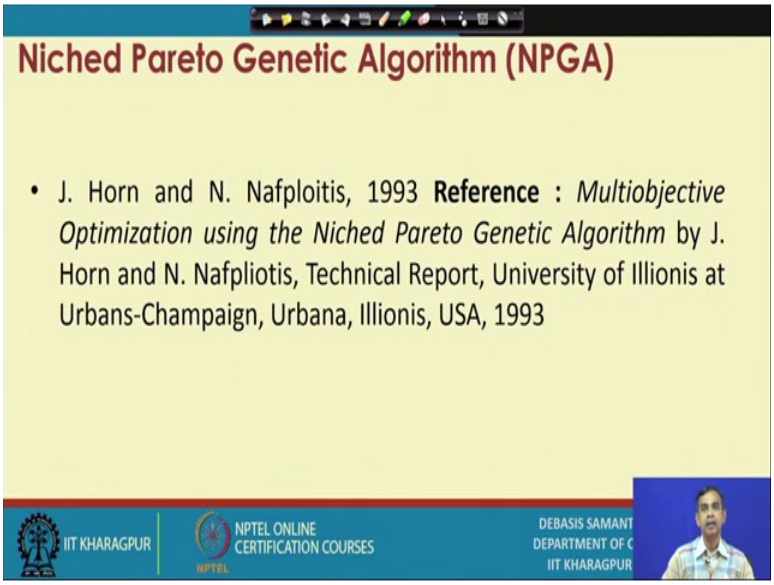
(Refer Slide Time: 00:50)



So, this algorithm is basically based on the rank plus niche calculation. Now we have learned about how to calculate the rank here we will calculate the niche count niche count essentially calculate in fact, how a solution is preferable than the other solution so, far the dominating solution is concerned.

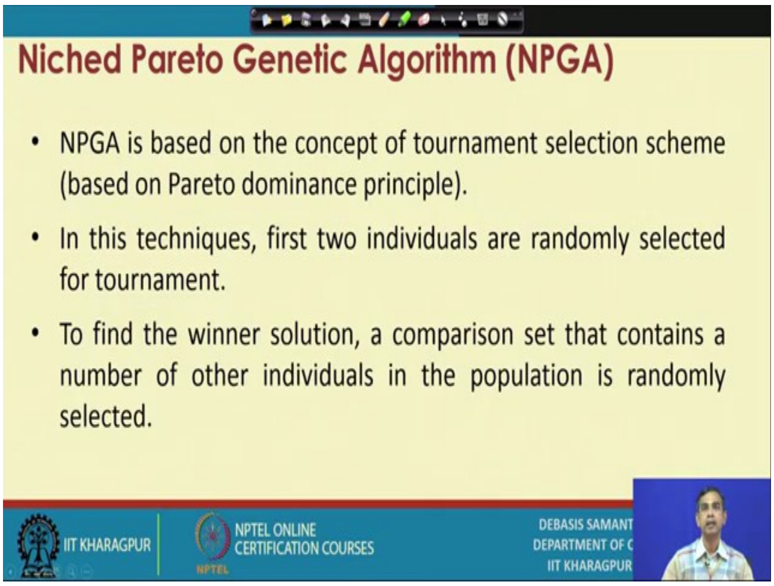
So, this is the second approach second Pareto based approaches to solve multi objective optimization problem and this is called the Pareto based because it will give a number of solutions which are of Pareto found. So, it is the Niched Pareto Genetic Algorithm.

(Refer Slide Time: 01:35)



This is the algorithm first time reported in the year 1993 and the research paper which was published in the form of a technical report by university of Illionis at Urbans- Champaign USA, and the work proposed by J. Horn and Nafploitis the title of the work is multi objective optimization using the Niched Pareto genetic algorithm. So, that is why it is called the NPGA Niched Pareto genetic algorithm.

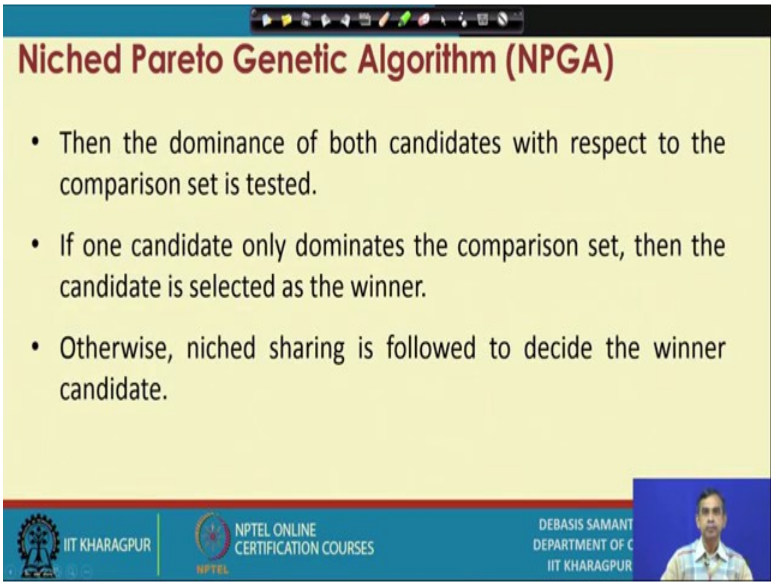
(Refer Slide Time: 02:13)



Now, the basic concept that this algorithm follows is the concept of tournament selection. I told you, all the multi objective optimization problem solving basically find a different selection strategy, so that we can select the non-dominated solution. So, here in this approach the selection that it follows it basically based on the tournament selection concept. So it is based on, the tournament selection is based on Pareto dominance concept. So, in this technique this technique is start like this in this technique we select any two solutions at random from the current generation and they are selected for the tournament.

And in the tournament there will be one winner. So, the winner will be selected and to select this winner in this approach it follows one set. It is called the comparison set. This set contains a number of solutions in the current generation and the number of solutions also again selected at random. So, it is a completely probabilistic selection strategy because the two solutions of which we have to decide the winner also decided at random and the comparison set, it is also a subset of the current population is selected at random.

(Refer Slide Time: 03:57)

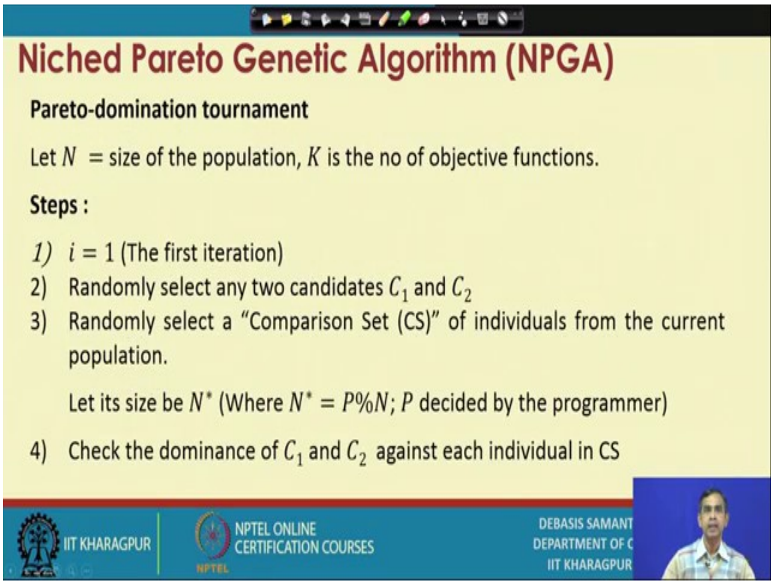


So, this is the concept of this technique. Now, so, first we have to select two candidate solutions at random and then these two candidate sets are tested at tested with reference to the comparison set that we have selected.

Tested in the sense that if the two candidates out of which if one candidate dominates all the solution which are in the comparison set then we can say that that candidate is the winner. Otherwise, if no one candidates dominate any one solution in the comparison set then we have to calculate one count of the two solutions. This is called the sharing count or also called the niched count.

So, based on this niche count the solutions which are having higher niche count will be selected for the winner. So, this way we can repeat the procedure for say N number of times say so that we can get the next generation and all the solutions in the next generation is basically the Pareto solution at the moment. So, this is the idea actually.

(Refer Slide Time: 05:19)



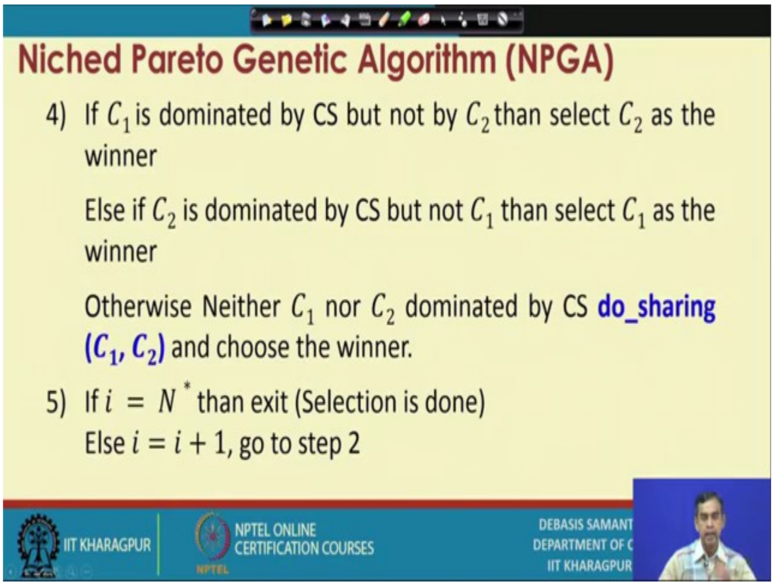
Now we can we can state this idea more formally in the form of a pseudo code and we will discuss this pseudo code here. So, suppose N , N is the size of the population and the problem which multi objective problem rather the multi objective problem which we are going to solve has K number of objective functions. Now, so let N be the size of the population and K is the number of objective functions are there in our multi objective optimization problem. So, it is basically an iterative step.

So, we will start with the first iteration. Let, $i=1$ it is basically tagged the number of iteration that it contains. Now the first step as I told you so that we have to select any two solutions randomly from the current population. Let, the two solutions are denoted as C_1 and C_2 they are the candidate solutions.

And then we have to select randomly a set. It is a subset of the current population. We denote this set as a comparison set and the size of the set is decided by the programmer. Let, the size of the set be $N^{¿}$ where $N^{¿}$ is P of the current P of the current population where the P is decided by the programmer. Usually it may be 10% or it is a 15% or 20% like this.

So, the depends on how fast you want to terminate it or how more exhaustively you want to search it depends on. So, if the P is very high then termination will be at a higher rate and if it is P is low, it is a slow algorithm, but it will be better solution at the time anyways. So, so $N^{¿}$ be the size of the comparison set at any moment. Then the next task is basically here is the tournaments selection should be applied here. So, we have to check the dominance of C_1 and C_2 against all the solutions are there in the comparison set. So, now so check the dominance, now let us see how the dominance can be checked.

(Refer Slide Time: 07:55)



It is basically so here so dominance solution is basically following the concept of domination that we know that mean with respect to so there are three two criteria could do conditions rather if C_1 satisfy all the conditions with respect to any solutions which belongs to the comparison set then we can say that C_1 dominates any solutions there in the CS .

So, if the solutions is C_1 is dominated by CS , but not by C_2 that means, C_1 is not inferior whereas, C_2 is inferior then select C_2 as the winner. So, in that case C_2 because C_1 is dominated by one solution. So, C_1 is not a good one. So, in that case a solute takes the solution C_2 on the other hand if C_2 is dominated by CS , but not C_1 then select C_1 as the winner.

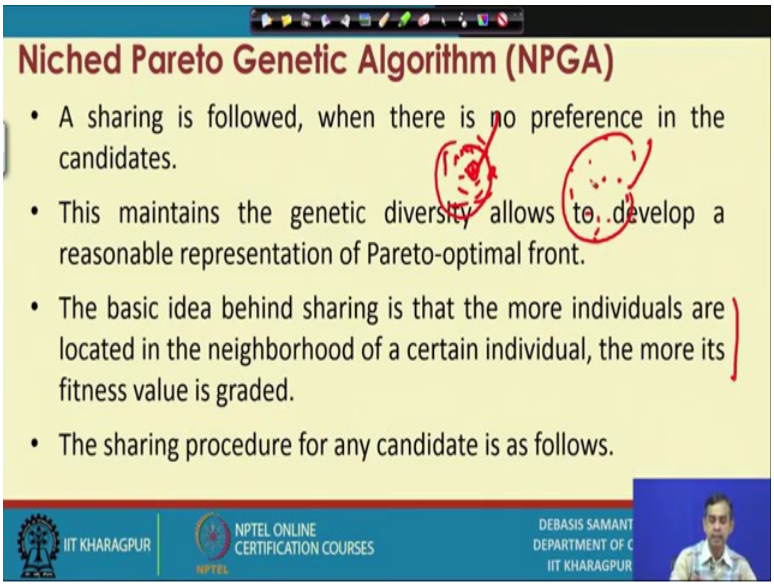
So, the two conditions if anyone is dominating then that can be selected as the winner. Now, there may be one case where neither C_1 nor C_2 dominated by CS .

So, in this case in this case we have to follow one procedure it is called the sharing count calculation. So, we term this procedure as do sharing between C_1 and C_2. Now do sharing procedure I will discuss in details later on so this based on this do sharing procedure it will return either C_1 or C_2 as the winner and then we will select finally, the winners from there. Now this step the step that we have discussed right now has to be repeated until ok.

We can select $N^{¿}$ until we select all the so it is basically until you select $N^{'}, N^{'}$ is basically size of the mating pool. So, here N' is also a parameter it is decided by the programmer; that means, the how so N' is basically denotes at any instant what will be the size of Pareto found like anyway.

So, it is basically we can consider N' is the size of the mating pool that needs to be considered to generate the population at the next generation. So, this is a repeative or iterative step we have to follow it and then we will find N' number of solutions for the mating pool. Now so, this now we will discuss about the method what is called the do sharing method.

(Refer Slide Time: 10:44)



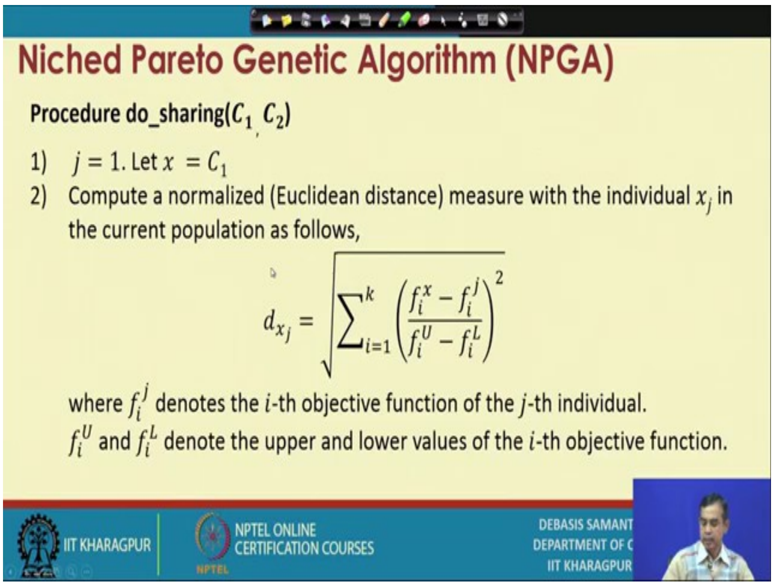
Now, so, here basically do sharing is a process of sharing count it is called the also niche count so and so basically we have to follow this sharing procedure when we see that out of the randomly selected two candidates no one is the winner and the best the main idea about this do sharing calculation is to maintain a good population diversity which allows to develop a reasonable representation of Pareto optimal front.

So, it is basically the idea about this one; that means, how the population diversity can be maintained. Now here the basic idea behind the sharing is that the more individuals are located in the neighborhood of a certain individual the more its fitness value is regarded so that is why the thing is that.

So, basic idea is that if this is the one solution and it has a number of solutions near about this one. So, its niche count will be high on the other hand this is the one solution which has which has the only fewer solution then its niche count will be low.

So, depending on this if it has very high niche count then we will select this solution compared to this solution as a preferable so for the Pareto optimal solution because it represents many of the solution then which represented by this one. So, this way we will select the then this way we can count the sharing count or niche count and then we can select the winner solution from there. Now let us see how the niche count can be calculated and what is the method that is followed there.

(Refer Slide Time: 12:27)



So, here the idea it is that so this is the current solution and there is a all the other solutions which is there then you have to we have to find the niche count like so we have to find that from this solution what is the similarity from the other solution this similarity is measured by means of Euclidean norms the distance between this solution to this solution.

So distance between the two vectors we can say so this is the solution this current solution is the one solution and this is the another solution which is there is the another solution. So, distance from one solution to any other solution is basically a similarity measure by means of Euclidean norm.

So, the idea is that from the current solution right we have to calculate the distance from all other solutions which is there in the solution in the population. So, here idea it is like this so it it is again iterative process because we have to count the distance or distance from the current solution to this solution. Let we have to count the sharing count of C_1 .

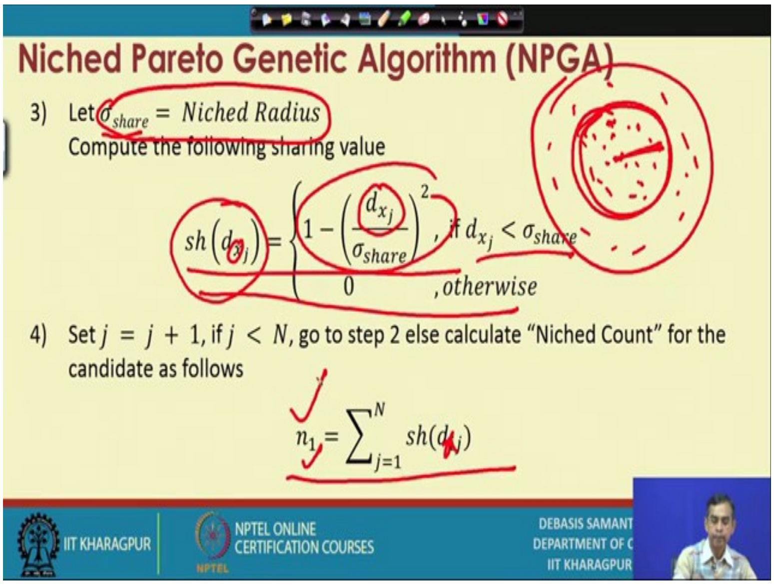
So, we will repeat it the same procedure will be repeated for the C_2 calculation also. So, this is the Euclidean distance that you have to calculate which basically measures the individual any one individual say x_j where j is starting with 1, x_1 and another is C_1 .

So, it is basically the distance between the currents x and the j-th solution. So, we denote it as a d_{x_j} . So, this is basically similarity between the two solution the current solution and any solution which is there. So, $j=1$ then we have to repeat for all other solutions so it is like this and for the $j=1$ any one solutions they are other than the C_1 .

Let see what is the idea about it. So, basically for each solution we have the objective function for the i-th vector. So, $i=1¿k$ so for each come function we have to calculate it is and this is the the i-th function value for the x solution and this is the i-th function value for the j-th solution and here also we consider f^U and f^L at the lower the upper limit and lower limit with respect to the i-th function.

So, once we know these things then we will be able to calculate it is the basically Euclidean distance form actually. So, it will it will return what is the similarity or the distance from the solution x to any j-th solution. So, so this way we will be able to calculate this d_{x_j} from the current solution to any other solutions which is there in the current population.

(Refer Slide Time: 15:27)



So, one the d_{x_j} is calculated then we will be able to compute the sharing count and to calculate the sharing count we have to decide one parameter. This parameter is called the σ_{share} parameter. σ_{share} parameter is also called the niched radius. Now this σ_{share} basically depends by it is decided it is determined by the programmer.

So, the idea it is like this so if this is the solution then it is basically the area out of which or within which we want to calculate the niche count so it is one sigma shear this is the radius like if we take the higher value then it is the higher one. So, the idea is that if we take lower value only few solutions are will be under confidence if we take the higher value then a large number of solutions will be under the confidence.

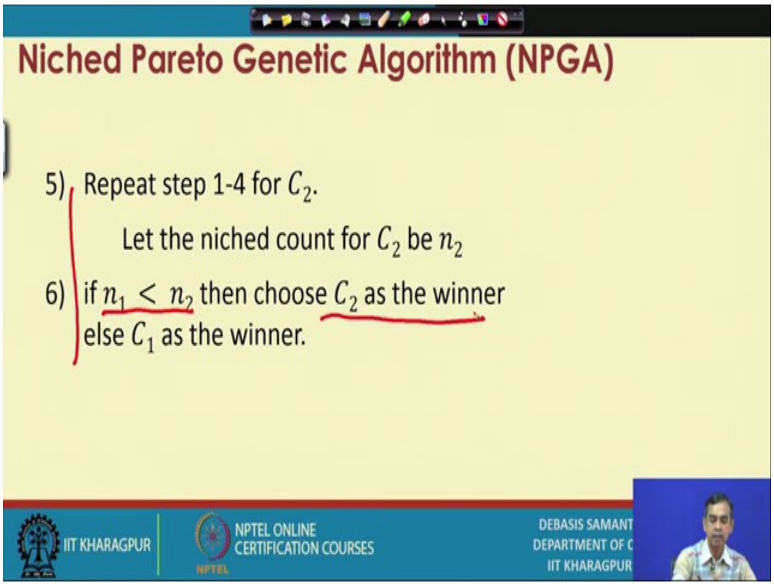
Obviously, it is here the cost of competition because for if it is smaller one then calculation is less. If it is larger one, then calculation is higher. Now once we decide the sigma share. So, this is the niched radius then we will be able to calculate the sharing count.

The formula for caring the size is shown here. If d_{x_j} with less than sigma share that mean if any solution within this radius, then thus it basically gives a contribution this contribution is like this one minus d_{x_j} we have already calculated earlier from the current solution x to any j-th solution and divided by σ_{share}. So, it is basically a normalized value and then so it is like this one.

And if it is 0 if any solution which is beyond or outside the niched radius, so it is 0. So, this way it will count only the sharing values or niched values within this solution value. So, this way the from the current solution x to any solution j we shall be able to calculate the sharing count or it is called the niche sharing count.

And then once we know the sharing count for all for with respect to the current solution and with other solution, we will be able to compute the niche count. So, niche count is denoted by n_1. It is basically summation of all the sharing count from the x with respect to the any other solution there. So, if $j=1 ¿ N$ then we will be able to calculate so d_{x_j} from x_j solution from the current solution to any other j-th solution. So, this basically gives you the niche count n_1 for the current solution C_1.

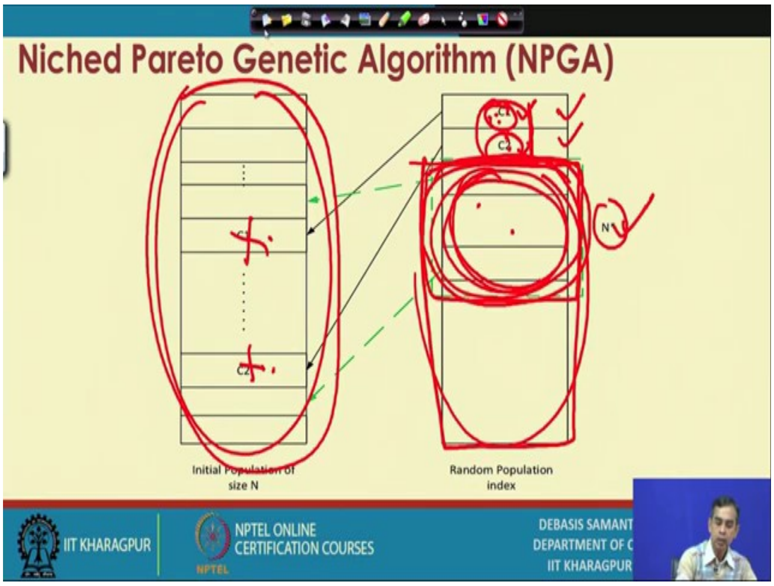
(Refer Slide Time: 18:20)



Now, the same procedure can be repeated for the candidate solution C_2 and we will be able to calculate another niched count let it be denoted by n_2. So, these are the two niched count n_1 for C_1 and n_2 for C_2. Then we can select the candidate based on this niche count which is shown here.

So, if n 1; if $n_1 < ¿$ n_2 then choose C_2 as the winner. Otherwise, we should we select C_1. So, this way if we select at random any two candidate solution then any one either one will be selected as a Pareto optimal solution, but this solution will be decided by the tournament selection. This is the idea about the tournament selection that it follows in case of niched Pareto genetic algorithm.

(Refer Slide Time: 19:11)



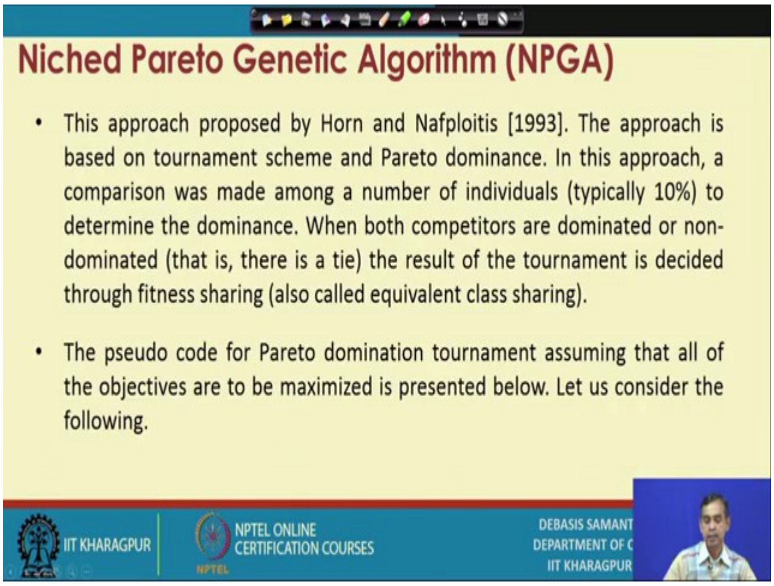
Now, here is the pictorial description to illustrate the same idea. So, this basically represents the current population at any instant at any iteration. Then we have to select at random any two solutions, C_1 and C_2 and these solutions are listed here. So, this is another random selection.

Next the P of this current population let this be $N^{¿}$ is selected. Then all the number of solutions which belongs to the number of solution number of $N^{¿}$ solution number of $N^{¿}$ solution will be selected from the current population except C_1 and C_2 because it has been selected that means, out of this things we have to select $N^{¿}$ number from the remaining and that also will be selected at random. So, it is in that sense completely random on.

So, let this is basically $N^{¿}$ number of subsets from this sets which is basically called the so this set is called the comparison set as we have discussed. Once, the comparison set is known then we have to see if C_1 and out of this C_2 if C_1 is dominated. So, if C_1 dominates all other solutions here in this comparison set. Then we can say C_1 is the winner.

Now, if suppose C_1 is not dominated by any solution then we have to check that whether C_2 is C_2 dominates any solutions are there. So, if C_2 dominates all the solutions which are there in the comparison set, but not C_1 then we will select C_2. Now if C_1 and C_2 neither one dominates any solution there, then we have to calculate the niched count for both C_1 and C_2 and based on this niched count we will select one solution. So, this is the idea about the NPGA algorithm the niched Pareto genetic algorithm and then it will calculate the solution.

(Refer Slide Time: 21:31)



And this step to be followed for N' number. N' is the number of mating pool right and from the this mating pool we will be able to produce the reproduction. The reproduction procedures and then make selection of mating pair are the same it is usual that is there in the conventional genetic algorithm.

Now so what we have learnt here is that then NPA the niched Pareto algorithm approach is basically is a different it follows a different selection strategy and this selection strategy is basically based on the concept of tournament selections. And in the tournament selection we consider the Pareto dominance concept.

So, it is basically Pareto dominance based tournament selection tournament scheme as the selection scheme in niched Pareto genetic algorithm. So, there are few conventions. So, the first convention is that in this approach we a comparison set was first considered or if generated with a number of solutions and typically it is 10% of the current populations solutions and when both competitors are dominated or non dominated so there may be tie then it basically the niche count this thing.

So, we have to resolve the tie actually; that means, if C_1 and C_2 neither dominated no one dominates the comparison set then we have to follow a tie and this tie is basically resolved by the niche count calculation. Now in the slides, I have mentioned some code, but this code is not so much important only the basic concept how it works that is important.

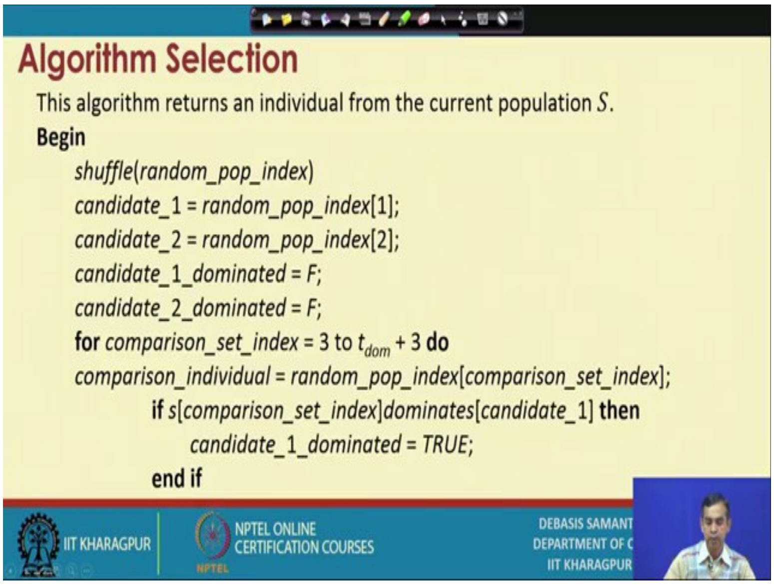
Now in this algorithm also we have to consider few more parameters σ_{share} so this is the one parameter should be chosen very carefully because if we select this parameter not so accurately or properly then it may leads to unwanted termination or termination with local optimum.

So, this risk is there and then; obviously, how much of the size of the comparison set also needs to be considered and usually this can be considered by means of empirical observation, empirical study. Then we have to start with different values of $N^{¿}$ value and then you can select it there is no other way of course, other than this one.

(Refer Slide Time: 24:12)

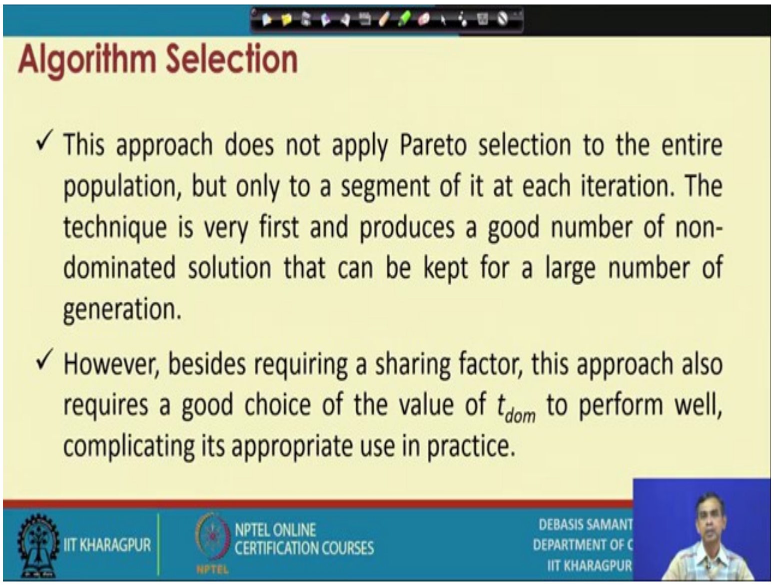
So, this is the idea about it and then that that is the code actually that code I do not want to discuss it here.

(Refer Slide Time: 24:18)



Code you can follow it and you can just understand the concept it is there.

(Refer Slide Time: 24:23)



Now, here so the selection algorithm is the main or critical one a step in this approach and here as we can note that this approach does not apply Pareto selection to the entire population. So, this is the one what is called the criticism against this solution that it only considers the subset of solution and that is why I told you that there is an issue that how we can decide this subset.

Because if we do not decide this subset judiciously then also it may not give the good result or it may not terminate in a finite time. So this is the one what is called the criticism that this solution is apart from that it does not consider Pareto selection to this. This means that it has certain chance that you will trap into the local optima.

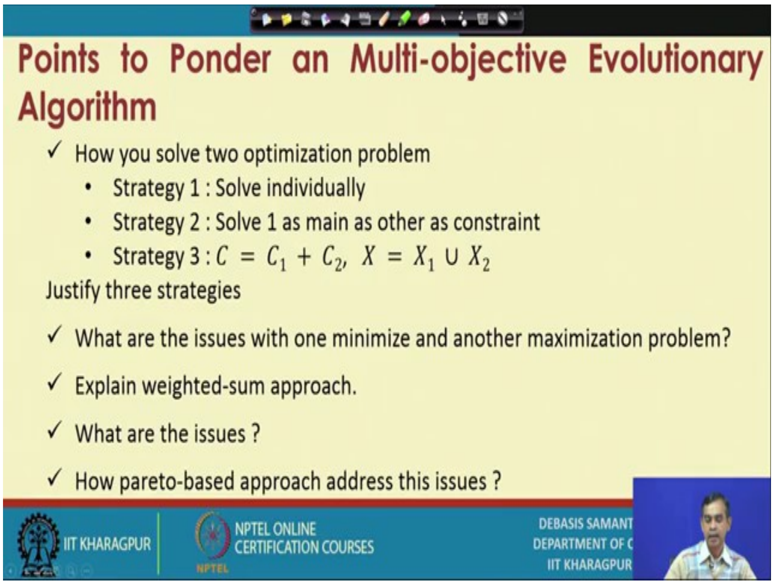
However, these technique is very fast and produce a large number of non-dominated solution that can be kept for a large number of generation. So, this basically can be considered there are many fine I will discuss another one idea about it. So, idea is that sometimes we have to consider two or more approaches to solve the multi objective that is called the hybrid approach.

For example, using so this approach NPGA as it is first in compared to the other Pareto based solution. So, we can use it to generate one solution which are basically solutions or the Pareto front and then another solution another approach which are may be relatively little bit high in computation demand can be followed but only based on the Pareto optimal form that we have discussed here.

So, in that in this case the NPGA can be considered with a very large comparison set large means not necessary 10% it is so maybe say 75% and then the comparison then we can select the Pareto front and then based on this Pareto front as the solutions for the genetic algorithm so you can follow some other multi objective optimization problem so approach solving a MOOP approach like say MOGA and then we can select finally, the Pareto optimal front.

So, this is the idea that can be followed. Basically it is a hybrid approach means two different approaches can be followed that approach can be both of the approach can be Pareto based approach or one may be non-Pareto based approach and another Pareto approach like this.

(Refer Slide Time: 27:19)



So, this is the thing if your time is possible then we will be able to adopt these things otherwise it is very difficult to follow this one. Now before conclusion just I want to have some discussion about it as I told you hybrid approach like. So, there are many strategies one is that non-Pareto based approach what we have discussed earlier.

So, solve individually; that means, maybe say VEGA approach or maybe say some other approach or SOEA approach is like this then or may be lexicographic ordering also you can consider and then we can solve one as the main and other is the constant like.

So, it is like it is there in the approach that we have discussed in lexicographic ordering actually and then we can follow the two solution combine and then the resultant solution can be obtained. And also we have understood about if one solution is to be minimized and other to be maximized.

How they can be converted uniformly all the minimization problem or maximization problem. It is not an issue because we have to calculate the dominance relation between the two solution they are they can be all minimized or can be a mixed type minimization and maximization, but that condition can be checked simple a programming solution is required.

And also we have discussed about weighted sum approach that is they are in the SOEA and we understood that this SOEA is very fast and then no need to do anything else only to calculate some a priori knowledge like what is the weight values for the all the functions there and then that also can be followed and then the solution that can be obtained from the SOEA approach can be used for the other Pareto based approach that is a hybrid combination of this one.

And Pareto based approach are the best solution compared to the non-Pareto based approach. However, Pareto based approach are computationally expensive compared to the non-Pareto based approach. So, if the computation time needs to be adjusted then we have to think some other strategy there in the solution ok. So, this is the algorithm Pareto based algorithm that we have discussed about niched Pareto based genetic algorithm and there are two the most advanced Pareto based solution this is called the NSGA algorithm and NSGA 2 also there. We will discuss all this thing in the next slide.

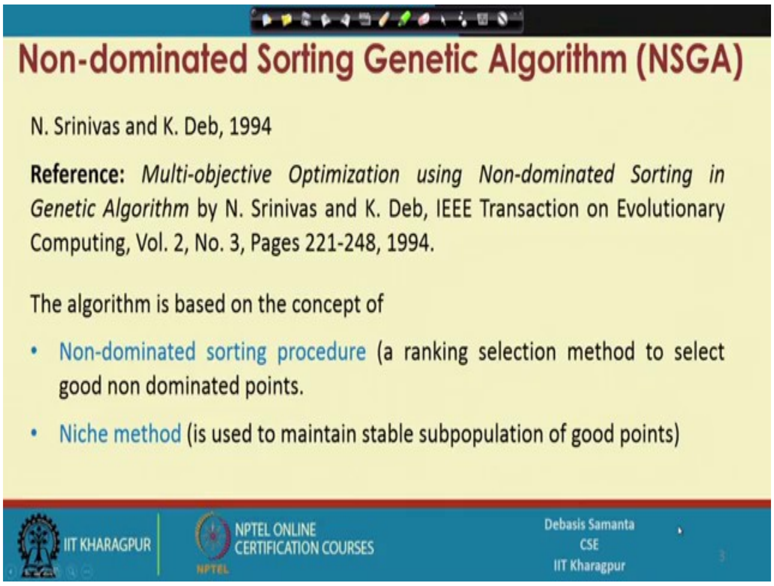
Thank you very much.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 31
Pareto- based approaches to solve MOOPs

So, we are learning about different approaches to solve multi objective optimization problem. In the last few lectures, we have learned different approaches belong to non pareto based and pareto based approach. We will continue our discussion today we will learn few more pareto based approaches. So, first we learn about the most popular one compared to the other pareto based approach it is called non-dominated sorting genetic algorithm shortly it is termed as NSGA.

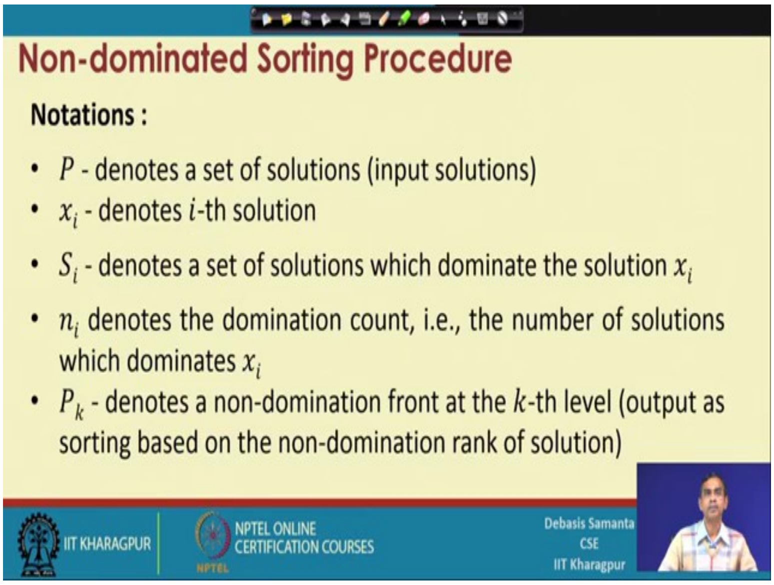
(Refer Slide Time: 01:00)



So, today learn about this the technique the NSGA algorithm, this algorithm first time proposed by N. Shrinivas and K. Deb in 1994, they have published this work in the journal called IEEE transactions on evolutionary computing, and the title of the paper was Multi-objective Optimization using Non-dominated Sorting in Genetic Algorithm. Now, this algorithm is different from the previous algorithm in terms of the concept that they followed. So, it is basically based on the concepts of non-dominated sorting procedure, and then they used another method another concept it is called the Niche method.

So, they basically select the non-dominated front by means of non-dominated sorting techniques, it is basically a ranking based selection method to select the good solutions which are to be in the non-dominated fronts, and then in order to have a good population diversity, and then better selection pressure they follow while they assign the fitness value to the solutions, and they follow one concept it is called the Niched sharing. So, we will discuss about the two concepts, and then we will know exactly how NSGA the algorithm works.

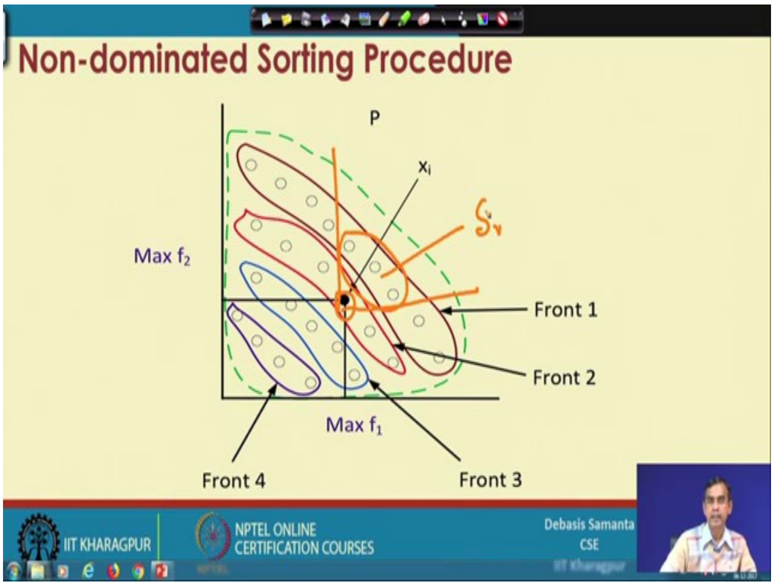
(Refer Slide Time: 02:43)



Now, in order to understand the algorithm, we will consider few terminologies. So, suppose P, P denotes a set of input solutions basically given the input solution. It is basically the current population we can say. So, we have to find for the current population all the solutions which are the, which are lying on the non-dominated front. So, P is the current solution. P is the current solution. So, P denotes the current solution and X_i , X_i denotes any i-th solution in the current set. So, X_i is a solution which belongs to the set P, we denote S_i . S_i is a set of solutions it is a basically subset of P which basically contains all the solutions which dominate the solution X_i .

So, S_i contains all the solution which dominates X_i , and then n_i denotes the domination count. It basically defines the number of solutions which dominates X_i , and we also denotes another notation called the P_k denotes a non-domination front at the k-th level anyway. So, we can explain all this terminology in with an example so, that we can understand about the concept.

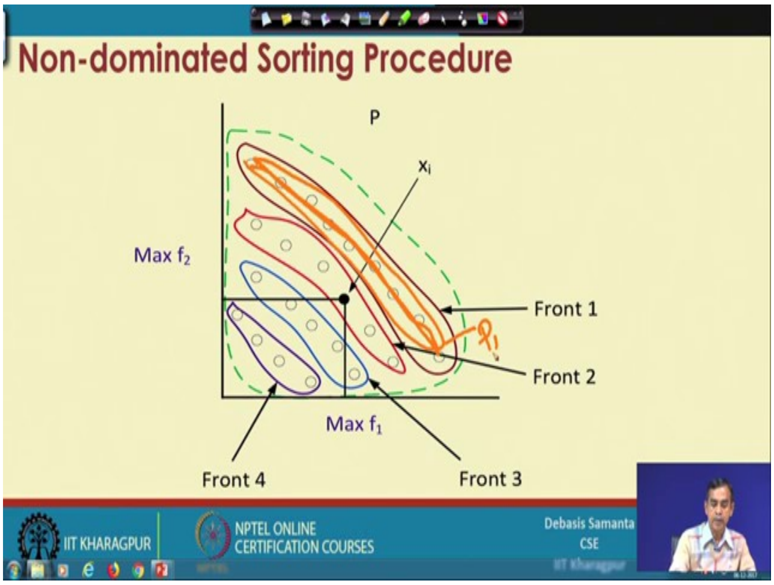
(Refer Slide Time: 04:37)



Let us follow this diagram. In this diagram. So, this basically denotes all the solutions, we say this is the P, P is the set of all solutions, and then we consider any solution X_i. So, if X_i and in this case we consider that f_1 is the function to be maximized and f_2 function to be maximized both the function to be maximized if. So, then this X_i is the solution, and then in this case the S_i who is denotes the solution who dominate X_i. So, in this case these are the solutions. So, these are the basically solution if this is this solution, then these are the solution basically called S_i these are the solutions which dominates this one, and then we defined the domination count n_i. So, basically by which all the solutions are basically this solution dominates all other solution in this region.

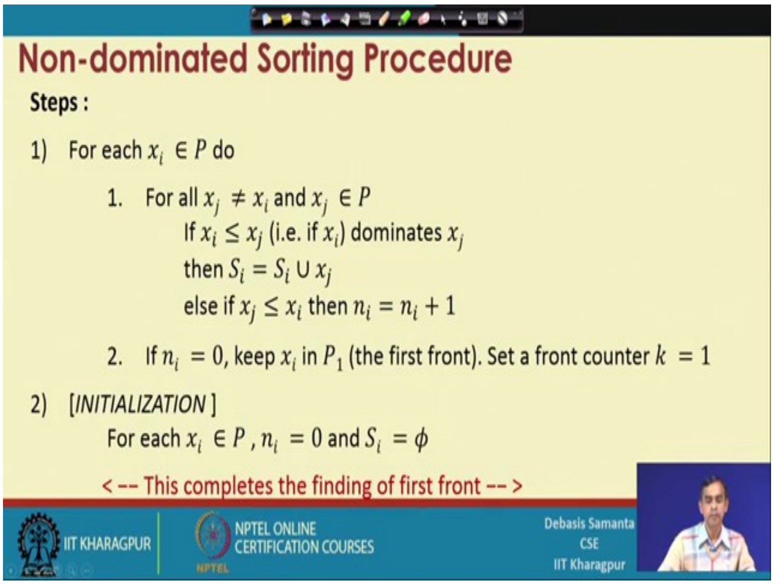
So, so the number of solution in this is basically n_i the solution counts of X_i, and the front. So, this is the solution which is basically the non-dominated solution because there is no solution which is dominated by this. So, these are the solutions is a non-dominated solution, and it basically creates a front we can say this front is 1.

(Refer Slide Time: 06:16)



So, it is basically this is the front $f_1(P)$ the first front now if we remove all the solution from this front, then it will give another front, the next front if we remove all these solutions from here, then the next front is basically next non-dominated front or next front. And similarly if we remove all this solution, then next front and next front. So, here the different front so Front 1, Front 2, Front 3, Front 4 like this one. So, these are the concept that is there. So, we have learned about the different terminologies. Now let us see how the algorithm can be defined here, and what is the procedure there in the algorithm.

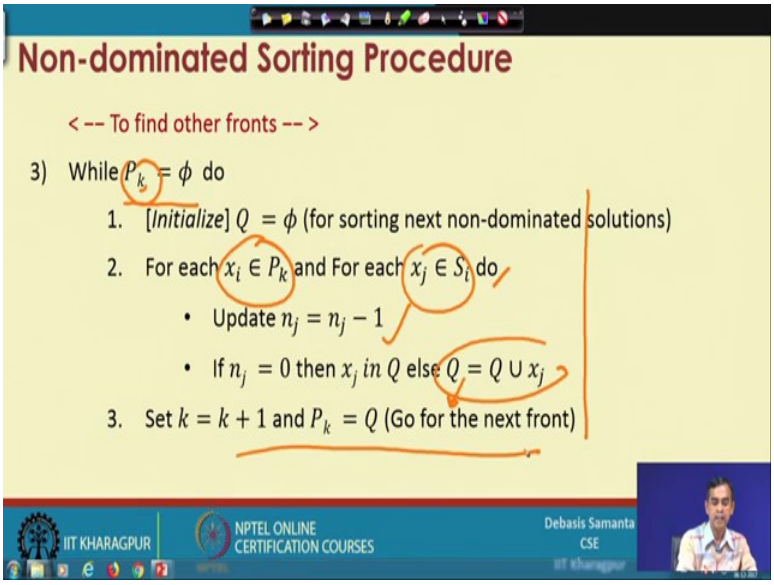
(Refer Slide Time: 06:55)



Now so, the algorithm basically finds the non-dominated front, and then the other dominated other fronts. So, let us see how this algorithm works. So, here this is the code that we have given it here this code is basically to find the non-dominated front that dominates outer most front, and here basically for each solution X_i which is in the set P, and for all other solution X_j such that it is not equals to the X_i and this X_j belongs to P then we will consider this one, if X_i is X_i dominates X_j, then we put X_j into the set S_i. So, S_i is basically the solution which basically dominated by X. So, it is basically if X_i dominates X_j when it is input in the solution S_i and if it is not dominated.

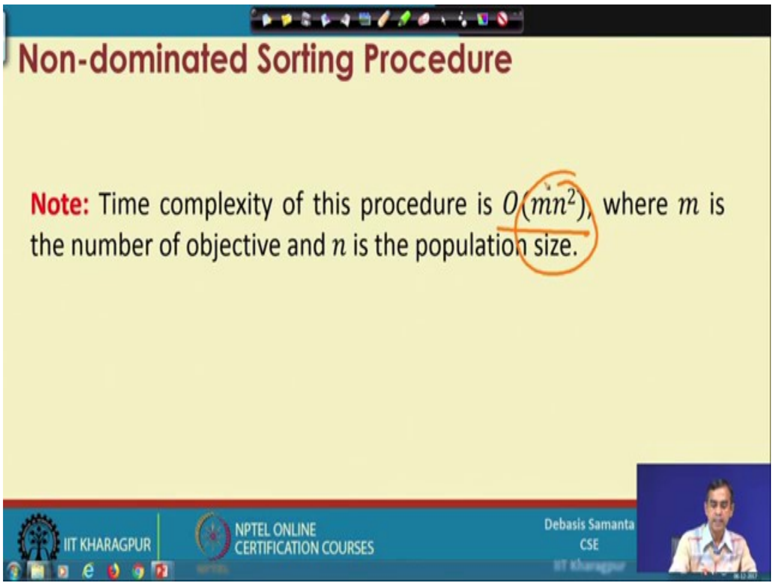
So, it is there; that means, it dominates X_j dominates X_i, then you can count the domination count. So, n_i equals to n_i+1. So, this way it basically counts the solution domination count as well as the solutions the dominated sets, and obviously, if $n_i=0$ for all the fronts that is there, then we can say this solution S_i is basically the P_1 that is mean the first front, and you can say k = 1. So, this basically the front and it is called the first front or non-dominated sorting front the first non-dominated sorting front.

(Refer Slide Time: 08:44)



And then we can find in the same way the other fronts. So, here the procedure so, it is basically initially say the k-th front, So, P_k initially it is phi because initially there is no element in the front then you if repeat this steps, then it will create the front P_k. So, for each solution X_i in P_k. P_k is basically the first it is it is start from the first front P_1, and for each solution X_j in S_i we have to do this one. And then we can add into the Q. So, finally Q will give the next front after the P_k front. So, this way we can find all the solutions all the other fronts in the solid state of solutions.

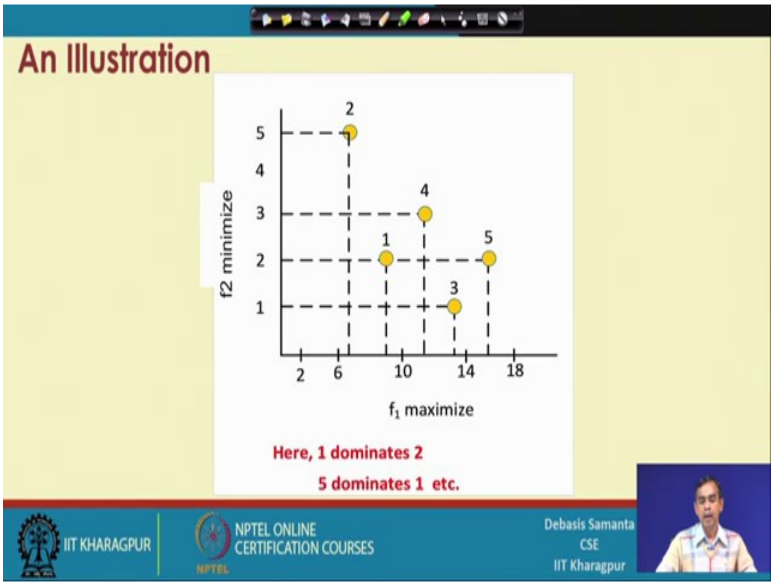
(Refer Slide Time: 09:35)



So, this way we can find is a little bit of programming approach, we can find using the concept of domination the first front, second front, and then n-th front depending on the number of solutions are here. Now so this programs I have mentioned this program in order to understand that how much time that it will take. So, it can be observed that the time complexity of this procedure of finding non-dominated sorting fronts, or it is basically called a non-dominated sorting procedure.

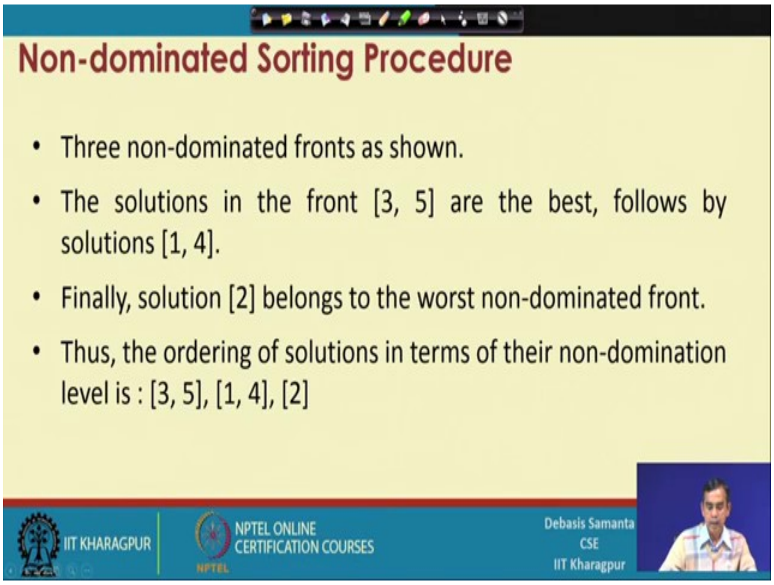
The time complexity of this procedure can be seen it $O(mn^2)$, where m is the number of objectives, and n is the size of the population. So, it is basically in terms of this complexity, and regarding this complexity, we will discuss few more things later on. So, this is the procedure the non-dominated sorting procedure is the main one critical procedure it is there in the NSGA algorithm.

(Refer Slide Time: 10:33)



Now, again I can illustrate this concept let us consider these are the solution set here, these are the solution sets here, and here f_1 is maximize and f_2 is minimize. So, if it is like this then we can easily understand that. So, this is these are the solution which are basically the first front, we can easily identify, and then this is the solution which is the second front, and this is the solution using the third front. So, in this concept there are three fronts, as we can say and the three fronts are like this three fronts are like this.

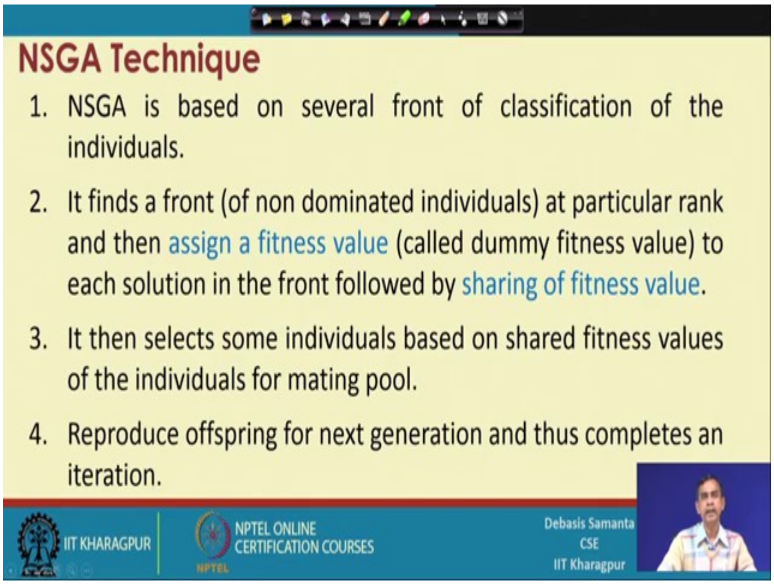
(Refer Slide Time: 11:23)



And so, here you can say that the solution in the front 3, 5 are the best solution, and they are basically called the non-dominated solution, on the other hand this solution 1 by 4 the next inferior solution, and the finally the solution 2 are the worst solution. So, for the front is concerned. So, these are the different fronts, and the different solutions are there. Now if we order all the solution based on their this front, then we can say these are the ordering.

So this is the first front, second front, next front, and so on so on. So, this ordering is called non-dominated so ordering right. So, what I want to emphasize is that a given a set of solution, we shall be able to find using the non-dominated sorting procedure, the different fronts and ordering of the different fronts, and so for the ordering is concerned. We can say all the solutions which belong to this front can be termed with rank 1, and this is the solution rank 2, and rank 3. So, non-dominated sorting procedure allow us to find the rank of all the solutions given a set of solution or current solution set. So, this is the one important concept that is fundamental, there in case of non-dominated sorting genetic algorithm.

(Refer Slide Time: 12:57)



And then we will discuss about the basic techniques which is followed there in NSGA algorithm. So, first we have already mentioned that it basically based on the classification of several front that can be there in the set of solution, and it finds in fact, it finds a front at particular rank, and then basically assign a fitness value.

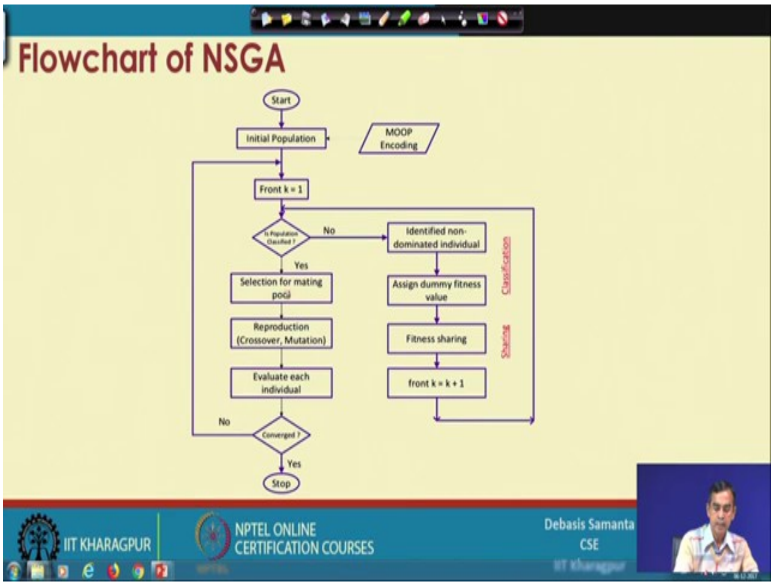
So, we will discuss about these are the concept that ok, we know exactly how to decide a rank of the solution this basically deciding the rank of the solution by virtue of calculate following the non-dominated sorting techniques, once the rank is known to us then we shall be able to assign a fitness value. So, this is one important task that is followed there, this fitness value is also called dummy fitness value to each solution in the front, and then the next technique that it follows is basically sharing of fitness values.

So, now our task is to understand how to assign a fitness value that is the dummy fitness value to each solution, and then how to share the fitness value, and what is the rational behind these things also we shall be able to learn. So, basically once this fitness value is assigned and sharing a fitness value, this means that each solution will be given a fitness value as per the NSGA technique. Once the fitness values are assigned to each solution then it basically considers the solution based on their fitness value to be considering the mating pool, and then they will be considered for the reproduction. Here, the reproduction procedure is same as the genetic algorithm.

So, basic idea is that we have to create the mating pool, and for this mating pool creation this is the selection strategy that it follows, and selection strategy means it basically find the non-dominated front, and then followed by that assigning the dummy fitness value.

And then finally the sharing fitness value. So, this basically gives the fitness calculation to all the solutions, and based on the fitness calculation we shall follow the reproduction.

(Refer Slide Time: 15:15)

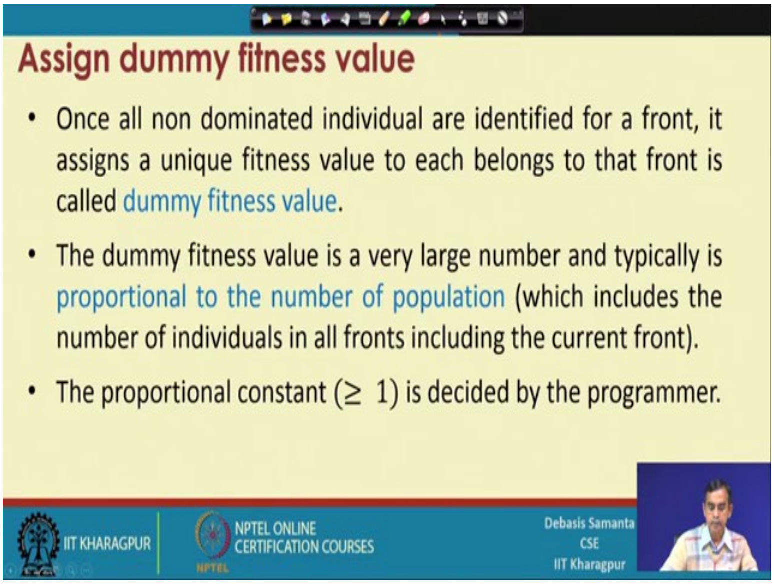


Now, so this diagram shows the flow chart of the algorithm. Now, if we little bit carefully check it. So, it is basically start with the initial population, and then it will basically consider the classification of the front, and then sharing of the fitness value assigning the dummy fitness value, this is the step which is there. Now here basically we start with the first front $k=1$, and then identify so $k=1$ basically we consider the first front, then because the k-th front and then assign dummy fitness values and dummy fitness sharing. So, this basically for the k-th fronts. So, k starting with 1 and this will procedure.

For all the solution until all the solutions are assigned the fitness value and sharing this one. Once all the solutions are assigned some fitness value, then it will come here. It is basically selection of mating pool. The selection can be the same selection that is there in GA technique like roulette wheel selection. Once the mating pool is selected then the reproduction procedure, and then evaluation of the solutions, and if we check the convergence criteria reached or not if reached this is the solution that can be obtained here, otherwise it will repeat the procedure for the next generation. So, the idea it is like this and so, basically if we see this is the conventional GA framework whereas, these basically the idea about how the selection techniques is different than the conventional GA technique.

So, this is the flow chart of the algorithm NSGA. So, there are mainly two tasks classification and sharing, and then we will discuss about the sharing concept how the sharing is there, and then sharing is basically use the fit dummy fitness value assignment. Now let us see how the sharing and dummy fitness value assignment is there in the NSGA 2.

(Refer Slide Time: 17:14)



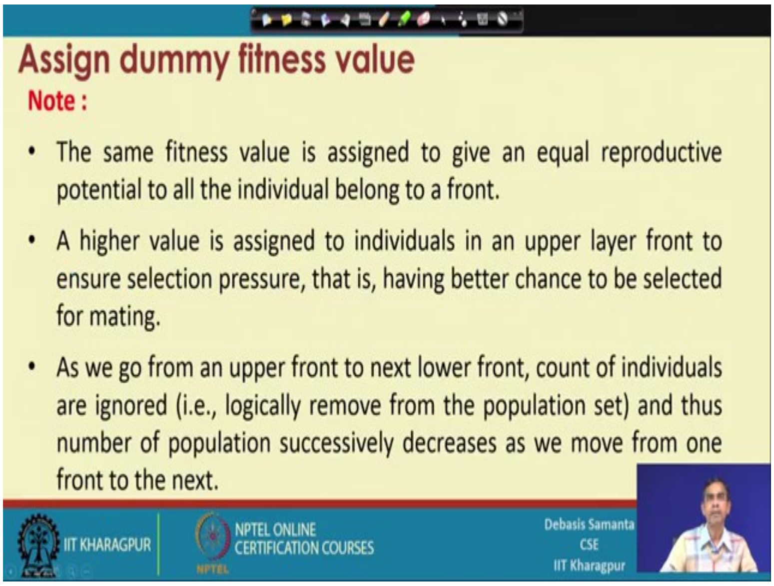
So, here the basic idea about assigning dummy fitness values is basically we first find all the fronts. Then assign dummy fitness value to each solution belongs to a particular front. So, here basically the idea it is that dummy fitness value is a very large number, and typically it is proportional to the number of population, which includes the number of individual in all front including the current front. Now here is the idea it is that at an instant if this is the one solution.

We have to assign a dummy fitness value it basically assigns one number which is very large number the number is denoted by all the solution which is in other which is not in the which is in this front as well as all the fronts below this front. So, it is the concept all the solutions, and then plus all the solution in this for include this is the large number of values are there and this assign.

So, it is again you can see all the solution which is there in the front is basically assigned the same dummy fitness value. So, this way we can assign the dummy fitness value, and it is a proportional number sometime proportional constant will be ≥ 1 to have a very good number. So, this basically assign the dummy fitness values to all the solution, and you can say that if we assign the dummy fitness value to the non-dominated front; that means, the first front they have the very higher number than the next front and so on.

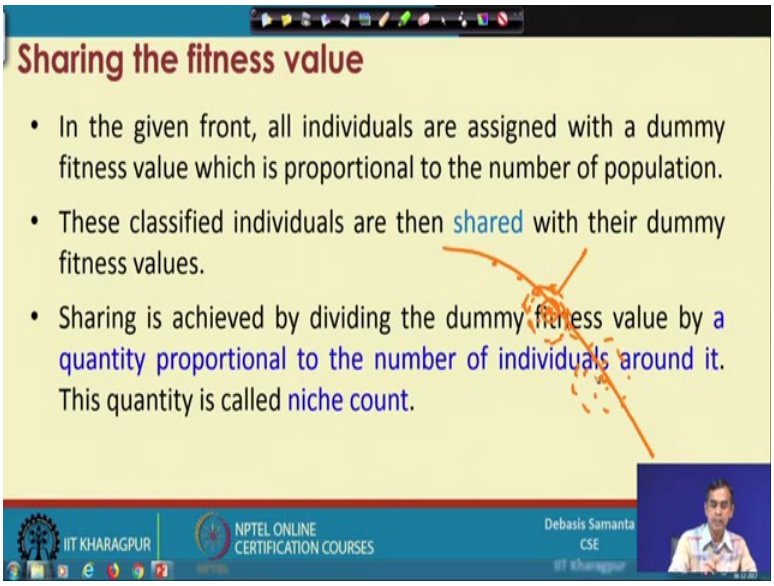
So, on as the front goes higher the dummy fitness value assign goes lower. So, this way we can basically give the fitness values is a tentative fitness values now. So, that can be very large and for the best solution for the superior solution, has the large number of what is called the dummy fitness value is compared to the inferior solution to that.

(Refer Slide Time: 19:19)



So, this basically assigning the dummy fitness values. So, here actually we can summarize the things that is there. The same fitness value is assigned to give an equal reproductive potential to all the individuals belong to a front. A higher value is assigned to individuals in an upper layer front to ensure selection pressure, that is, having better chance to be selected for mating. And, as you go from an upper front to the next lower front count of individuals are ignored that is logically removed from the population set, and thus number of population successively decreases as we move from one front to the next front ok. So, this is the idea about dummy fitness value assignment.

(Refer Slide Time: 19:59)

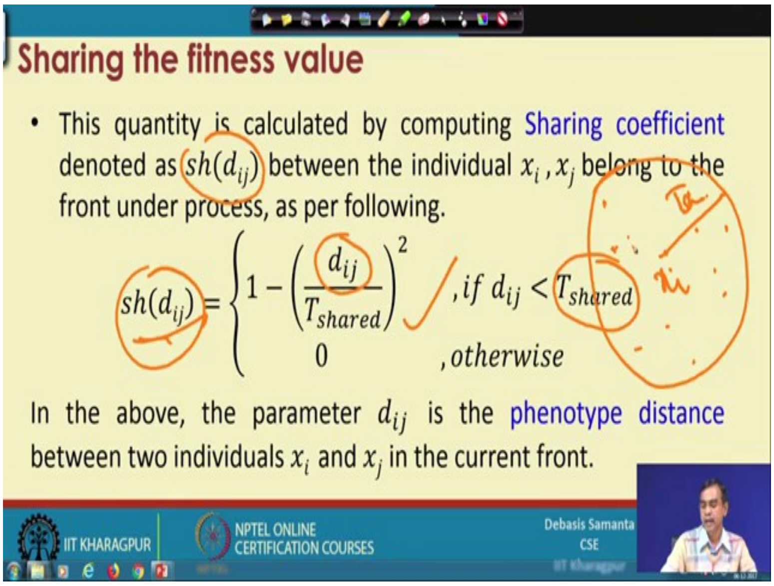


And our next concept there in the NSGA sharing the fitness value. So, here basically we have understood that assigning dummy fitness value, the main purpose is to have a very good selection pressure, and then the sharing the fitness value the objective is to that how the how population diversity can be there. So, that we can find better solution and the solution cannot be tapped into the local optima.

So, here the idea about sharing the fitness value is like this. So, in the in any front all individuals are assigned with a dummy fitness value which is proportional number of population. These individuals are then shared with their dummy fitness values. So, it is called the sharing concept. Now, sharing is divided a sharing can be of sharing is achieved by dividing the dummy fitness value by a quantity proportional to the number of individuals around it.

Now this is the one concept. It is basically the niche concept. So, it is niche count. So, here the idea is that if so this is the one solution, and it one front, and we have already assigned one dummy fitness value. So, we have to see right we have to share the fitness value in the sense that. Now, we have to see what are the solutions which is around this one right. So, if this is the solution which has had a less solution around this one, then it has less niche count then this one. So, dummy fitness value that been assigned to this solution, if we divided by this assigned to this solution, if we divided by this niche count, then it is called the shared fitness value. So, the solutions which has very large niche count has the higher fitness value, and compared to the solution which has less niche count has the less shared fitness value. So, this way all the solutions when assigned the dummy fitness value, when a particular front has the same dummy fitness values after the sharing information they even all the solutions are same front have the different fitness values, and that is the ultimate fitness value that should be considered in the mating pool selection procedure. So, this is the idea about the concept niche count.

(Refer Slide Time: 22:15)

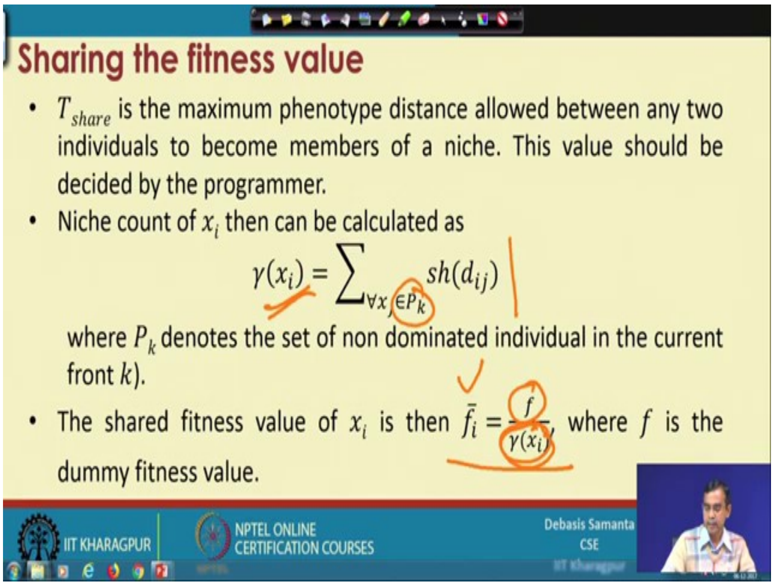


Now, so niche count how this niche count can be consider can be calculated. The idea it is the same idea once we have discussed about sharing the niche values in the NPGA algorithm, it is the same concept it is followed there in NSGA. So, here basically we say d_{ij} is basically the distance between two solution X_i and X_j, and again another constant that needs to be decided it is called the T_{shared}. So, it is basically if this is the solution X_i, and T_{shared} is denoted how far we have to consider that mean what will be the region of sharing basically. So, it is X_i and this is the radius it is called the T_{shared}.

Then whichever the solution which is there it will be considered the shared d_{ij} count $sh(d_{ij})$, and it is the formula this one. So, this is the same formula we have used there in NTGA algorithm. So, finally, for this solution and then for all other solution within this region it will calculate $sh(d_{ij})$, and then $sh(d_{ij})$ is basically the sharing information or it is basically the niche count. So, these are the niche count.

Now $sh(d_{ij})$ it is higher, if we find the number of solutions higher, and it is lower if the number of solutions this one. So, it is basically niche count; that means, it is basically how a solution is in a region which is basically having the higher population density compared to the others. So, the solution which has the higher population density has the higher value of niche count compared to the solution with the less population density. Now so, here basically the concept of how to niche count for each solution that can be obtained once this solution is there, then we will be able to consider about ok.

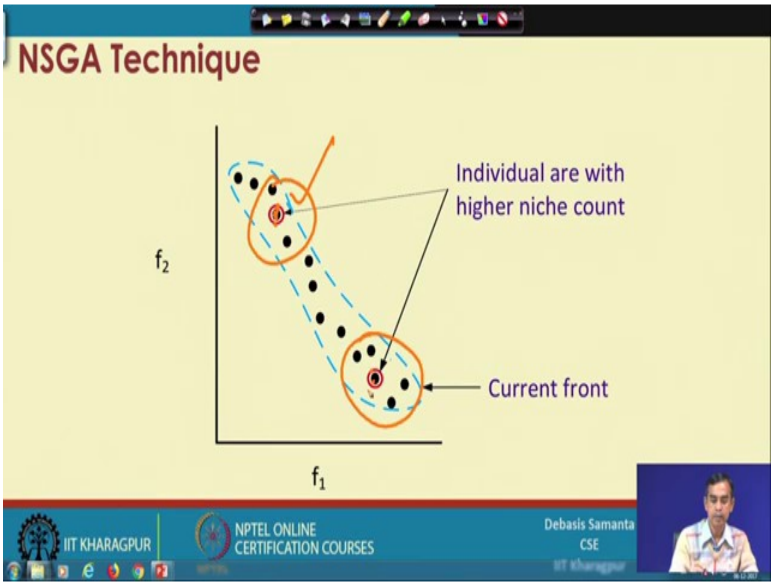
(Refer Slide Time: 24:09)



So, this basically the total niche count, if you consider all solutions with respect to the i-th solution belongs to particular front P_k, and then we can share the fitness value if f is the original fitness value of the i-th solution, and divided by the gamma X_i that is the niche count of the solution it is there. Now we can understand that the solution which has the lower population density has the fitness value it is this one higher fitness value, and the which has the niche count higher it has the lower value. Now it is little bit confusing like, actually it is the idea is that sometimes the good solutions should be combined with the bad solutions, this because population diversity is much more. So, if we consider all good solution then population diversity is less.

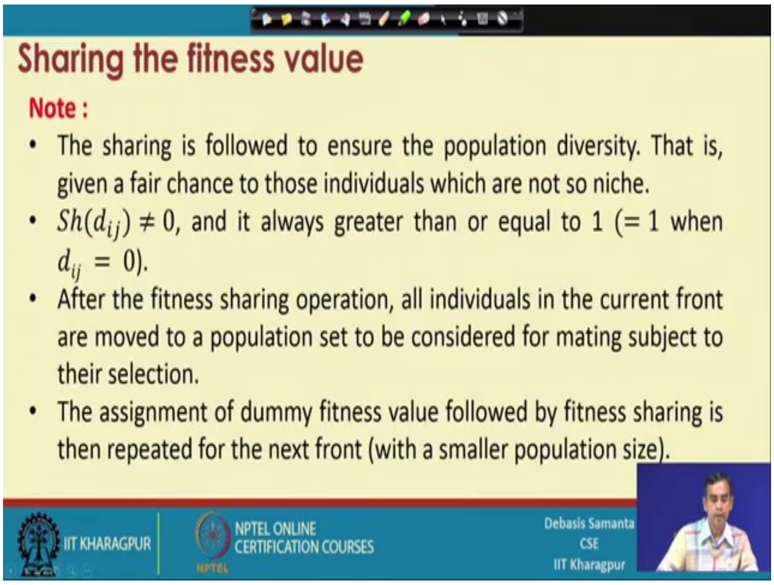
So, we have to share the fitness value. So, that the fitness value can be can be shared among the little bit solutions which are not so, good solution or inferior. So, share count is basically say that how a solution can be a good representative than the other solutions are there. So, this way we will be able to kind that this one. So, this basically the shared fitness value and these are final values that needs to be considered for all solutions before the mating pool creation.

(Refer Slide Time: 25:41)



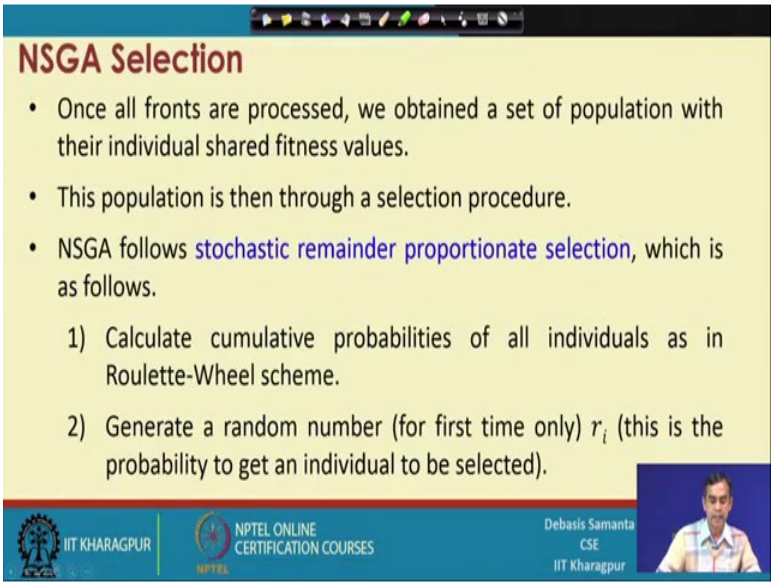
Now, so the idea it is like this so we have discussed about. So, this solution and this solution if this is this one, it has the niche count is much more then this solution has had the niche countless. So, we will select we will prefer for the mating pool compared to if their competitor, then we will prefer this solution than this one. All though all solution I have the same dummy fitness value, but whenever divided by this shared niche count, it is it gets more weightage than this one. So, this way we will be able to have maintain the population diversity.

(Refer Slide Time: 26:18)



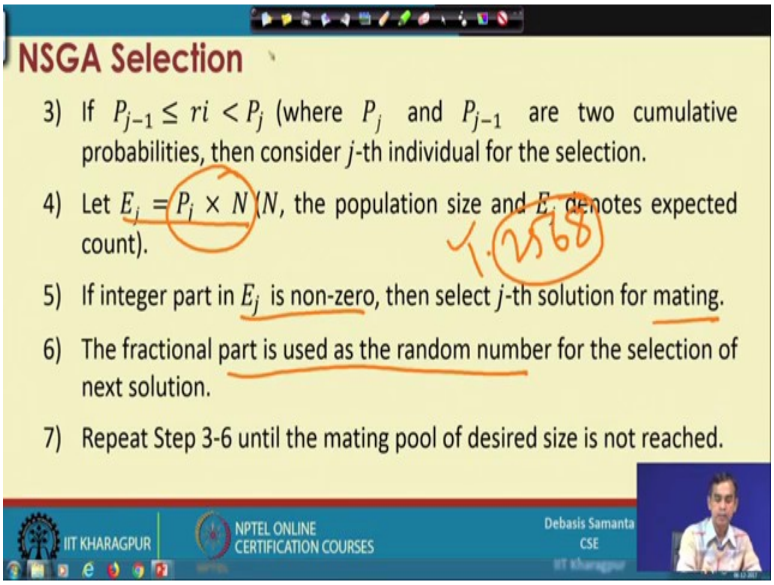
So, this way you can learn about how the solutions the particularly belongs to a particular front can be shared, and then we can consider for mating pole creation.

(Refer Slide Time: 26:31)



Now, so once it is there then we have to go for the selection procedure for the mating pool creation. So, all the solutions on the particular front, we have to follow one method called the stochastic remainder proportional selection technique which is discussed here, it is basically a roulette wheel selection technique, and generate basically random number r_i, and then follow the random number to consider the same procedure a roulette wheel in a cumulative probability is concept.

(Refer Slide Time: 27:03)



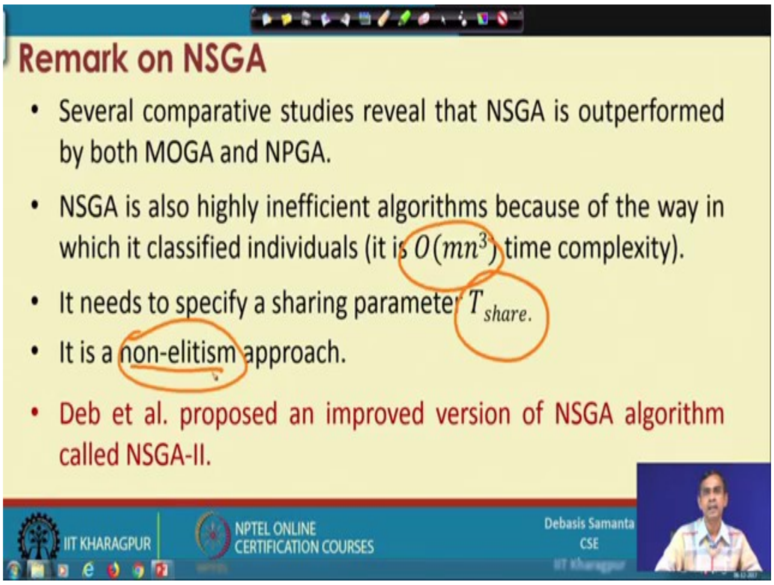
That can be consider and it can be so, used to select the solution there. So, this basically the same procedure that is there in the roulette wheel selection and, but here the one difference is that in this method is basically it calculates based on this r_i, and selection of the particular j-th solution, it calculates the expected count, we have learned about expected count while we have discussing about the roulette wheel selection method.

And then it basically the ok, in the previous roulette wheel method that we have learned, then we have to generate the random number each time, but it does not require to random generate number, in this procedure it basically follow the calculation of E_j which is using this formula, and then if E_j is non-zero, then the solution is selected, and then for remaining; that means, for the a non-integer parts we can use non 1.2568 so is basically this is the next random number to be considered if it is there. So, this is the procedure it is followed instead of generating random number it will follow, this concept so that we can avoid the compression there.

(Refer Slide Time: 28:13)

So, this way the selection can be done, and then mating can be for sure, and the reproduction procedure same as the reproduction procedure there in the simple genetic algorithm.

(Refer Slide Time: 28:22)



Now, I just want to conclude before going to stop, this discussion. Here, the so it has been observed that if we compare the NSGA technique with respect to MOGA, and NPGA it is observed that in most of the cases NSGA gives better result compared to the MOGA and NPGA. However, the main drawbacks of the NSGA is that this algorithm compared to the MOGA and NPGA is computationally expensive it has been observed that a computational time that is required for this is $O(mn^3)$ if you consider the entire procedures there.

So, other than this inefficiency so for that timing or the time that is the time that is required to compute this algorithm is concerned, there is another is that this algorithm should consider one parameter to be decided by the programmer, it is here in order to niche count. So, these are the two serious drawbacks, that is there in this NSGA and another approach is that this is basically non-elitism approach, because it gives a favor to some times the inferior solution also, because the non-dominated front is basically or the front we have considered they are basically the non-elitism approach.

So, these are the three, I mean critism against the NSGA, and then the same author who proposed NSGA, they have proposed another person of the NSGA, it is called the NSGA 2 algorithm. So, we will discuss this NSGA 2 algorithm in the next lecture.

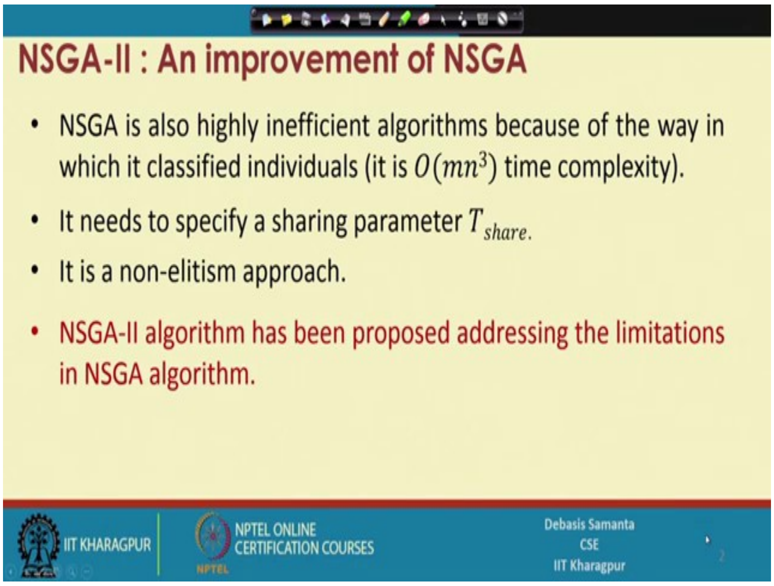
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 32
Pareto-based approach to solve MOOPs (Contd.)

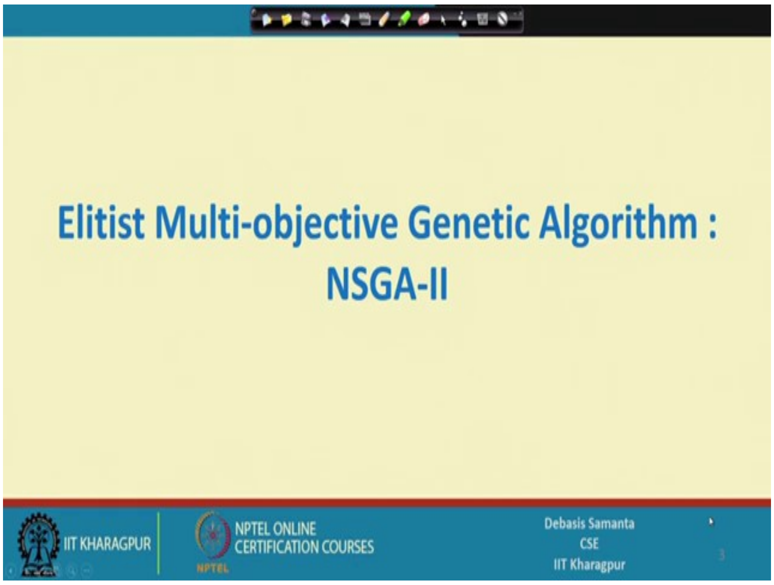
In the last lecture, we have learnt about NSGA non-dominated sorting genetic algorithm proposed by N. Srinivas and K. Deb. And we have discussed some limitations there addressing all those limitations the same authors with some other more researchers.

(Refer Slide Time: 00:38)



They have proposed the another algorithm, the next version, the improved version of NSGA, it is called the NSGA 2 algorithm.

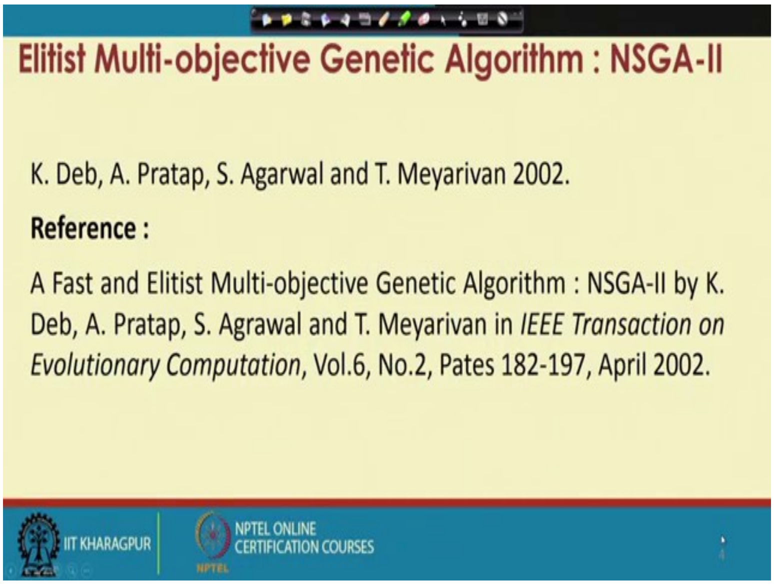
(Refer Slide Time: 00:48)



So, this algorithm is called Elitist multi-objective genetic algorithm and abbreviated as NSGA 2, and the basic concept in the NSGA or rationale in the NSGA 2 is basically so that it can be computationally much efficiencies, and then it gives better result compare with NSGA.

So, this is the idea about the here in NSGA 2 approach.

(Refer Slide Time: 01:19)

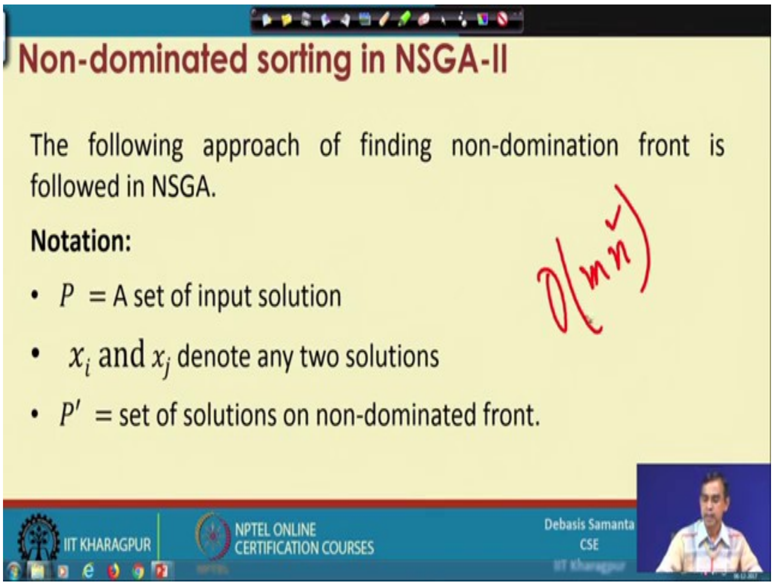


And here are the contributor who developed this algorithm, the contributors are K. Deb, A. Pratap, S. Agarwal and Meyarivan. This algorithm NSGA was first time proposed introduced in 1994, and around 8 years later this algorithm NSGA 2 has been proposed by the same group of researchers in the same lab. So, these lab is basically the lab they are in IIT Kanpur ok, and this lab is famous as NSGA lab.

Now, the work NSGA 2, first time in his published in IEEE transactions on evolutionary computation. And the title of the paper was A Fast and Elitist Multi-objective Genetic Algorithm NSGA 2. So, basic idea they claim that it is fast, and elitist. Now we will see exactly how the time complexity have been owned here, and then how the concept of elitists has been enjoyed here or is applied here rather, and the algorithm it is there as the algorithm is bit complex and lot of steps are there. So, we will discuss it in 2, I mean sessions, in the first time we will discussed the basic concept or overview of the things. And it considers some unique approaches there it is called the crowding sort. So, for the selection is concerned we will discuss in the next slides.

Now, let us start about the concept it is there in this algorithm.

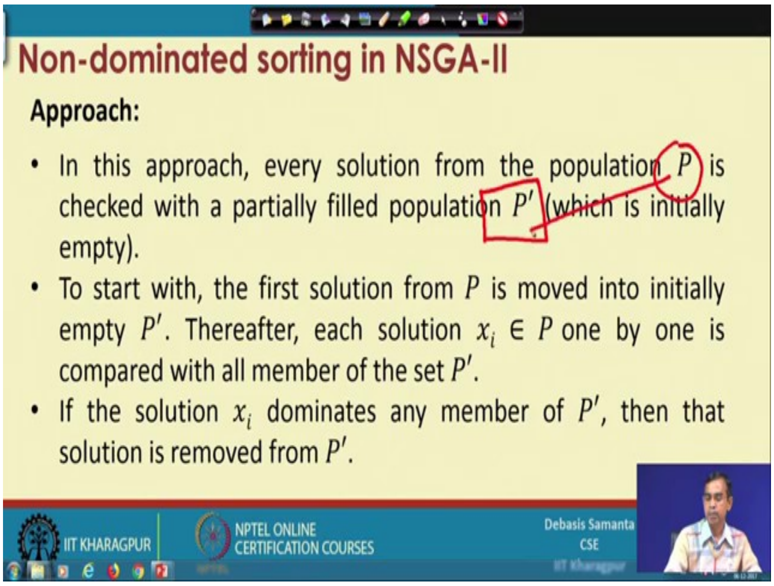
(Refer Slide Time: 02:54)



Now in order to understand, we can recall that the NSGA basically consider the that a front. They basically calculate all non-domination front it is called, and then once the front is calculated, then they assign fitness values to all the solution in a particular front this basically the dummy fitness values, and then sharing the fitness value by means of niche count.

However, the procedure that it follows it totally different. This is because finding the non-domination front itself is a computationally very much x expensive, we have discussed about that the complexity of finding the non-domination front is in the order of it is $O(mn^2)$, where m is the number of objective and n is the population size.

(Refer Slide Time: 03:40)



And we will discuss we will see that how the, it also considered the finding non-dimension front, but in a different way, we will discuss about this procedure.

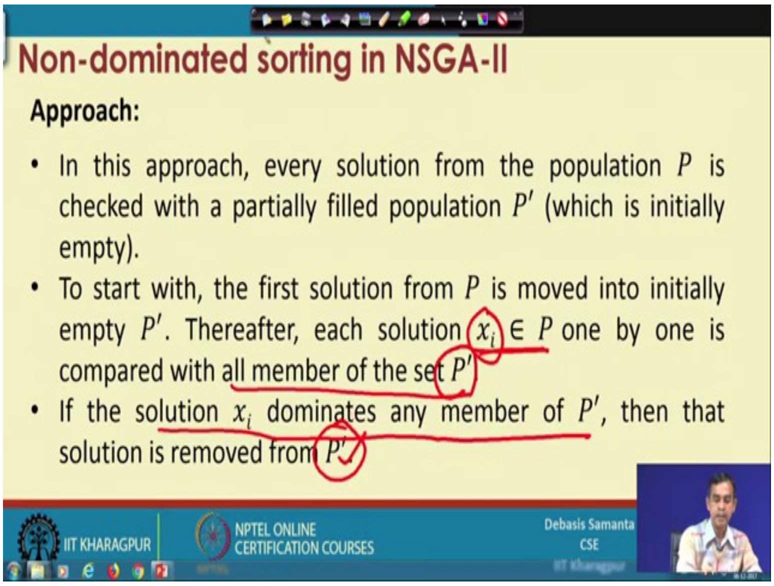
Let us first consider the different notation that it follows to discuss about it. So, like the NSGA procedure, we consider the P is basically input solution set; that means, set of all solution belongs to a current population. And we denote x_i and x_j are any two solutions in the set of solutions, and here P' is the set of solutions on non-dominated front. So, P' be there any solution it is there.

Now so, it is the, these are the terminal that will follow, and then the procedure the approaches is like this. So, here every solution from the current population said that is P, it checked with a partially filled population P' which is initially empty. So, the idea is that given a set of solution P, we have to first find a non-dominated front P'.

So, it is basically idea. So, how the from the set of solution P we can find P'? So, basic idea is that initially P' is empty, and we can randomly choose ok; so, basically initially P' is empty, and then we have to fill the P' with some solutions they are basically the solutions are in the domination front. So, P' basically say contains all the solutions belong to the domination front at any instance.

Now, to start with this; that means, filling the P' from the P what basically idea is that we have to randomly choose on solution x_i which is in the P one by one is basically; that means, all the solutions should be checked one by one which is there in the P.

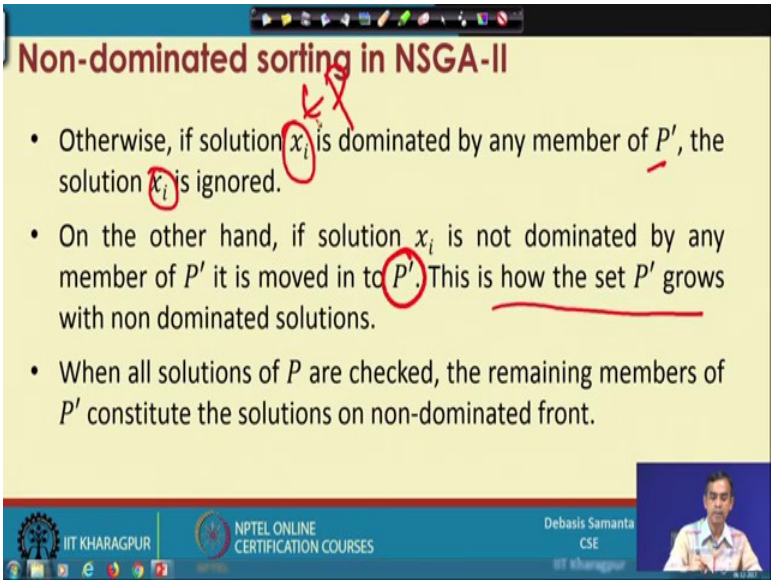
(Refer Slide Time: 05:42)



Now here so, so, compared with then this x_i is compared with all members of the set P'. So, initially the P' is empty; this means that, x_i should be there in the P' initially. Or in other words, suppose P' is not empty, then it basically check this on. If x_i dominates, any member of P' then that solution is removed from the P'. The because the P' should contents all solution which is not dominated by any other solution. That is why P' should be a non-domination front.

So, it is removed there and the same procedure is repeated for all other solutions one by one, and then P' will be gradually filled, and then P' contents all the solution in the non-domination front.

(Refer Slide Time: 06:40)



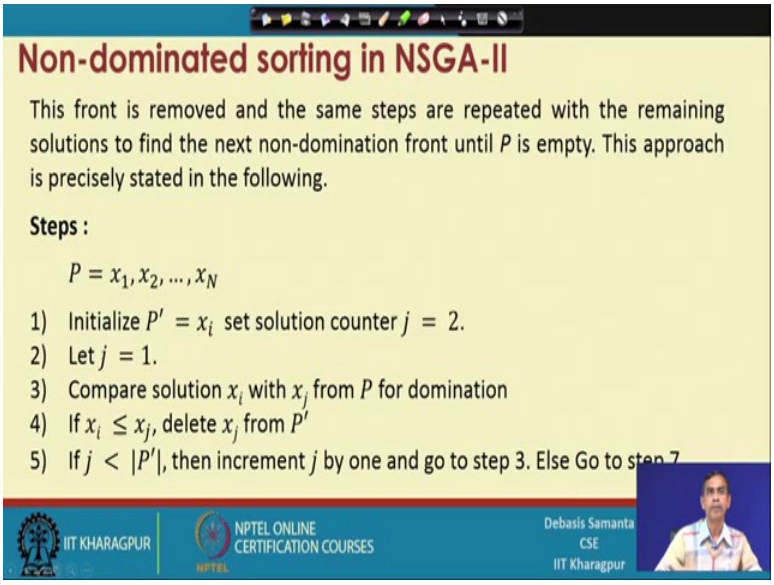
So, this is the idea about this one. So, here again repeated is if solution x_i is dominated by any member of P', the solution x_i is ignored. So, the solution needs to be need not be consider in the P'.

On the other hand, if solution x_i is not dominated by any member of P' it is included into the P', because x_i is therefore, the solution to be in the non-dominated form. So, this way the P' grows when we checked all the solution x_i belongs to P the current solution one by one. And when all the solutions of P are checked, then the remaining members of P' constitute the solution on non-dominated front.

Now, you can understand this basically the technique in the last slide and these slides includes and this basically the clever idea about the previous approach. So, this solution is better this solution is basically find the non-domination front P'.

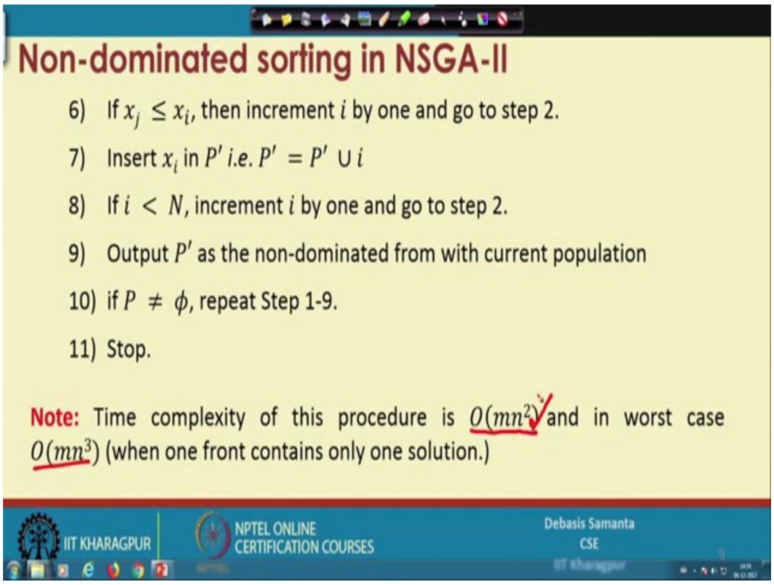
Once the P' one, first front is obtain you will remove all the solution from the P', and repeat the same procedure finding the next front and so on. So, this method is therefore, different than the non-dominated sorting procedure in NSGA. And it is faster than the procedure that is there in NSGA. So, this way it is a first method of finding non-dominated sorting front.

(Refer Slide Time: 08:07)



Now so, the idea it is like this so, so, it is basically it will these are the steps that we can follow, and more precise steps that is there in the calculation of non-dominated front there.

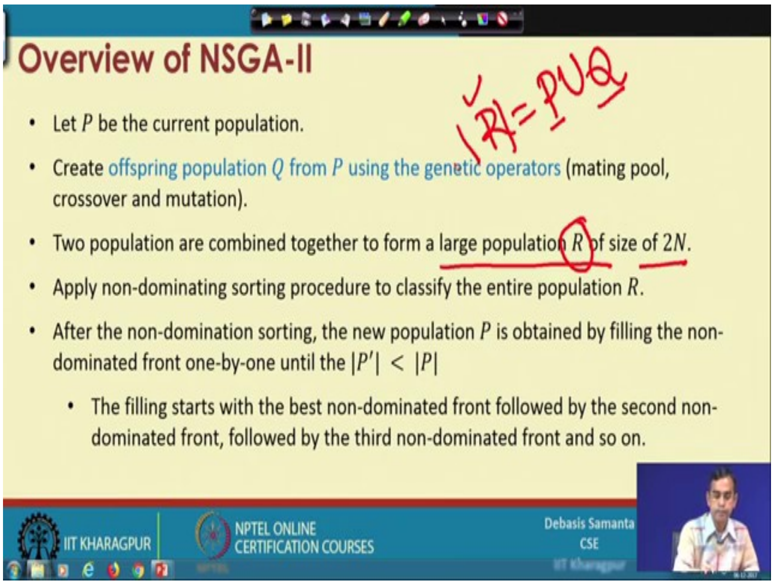
(Refer Slide Time: 08:22)



And we can observe that a detail calculation can let you that the complexity of this procedure is $O(mn^2)$, and in worst case it is $O(mn^3)$, although the same complexity here, but still it gives the better what is called the time than the previous one.

So, so this way this is the improvement over the NSGA so far, the non-dominated sorting front calculation is conserved.

(Refer Slide Time: 08:58)



Now we will discuss about the basic approaches of NSGA 2. First the thing is that how to find a non-dominated sorting front. Now we have learned about that complexity of the two algorithms are same. So, anyway both NSGA and NSGA 2 needs to calculate all the front it is there. So, whether the new method that we have discussed just now can be followed or the method that is there in NSGA also can be followed in order to calculate the non-domination front anyway. So, non-domination front calculation is there in both the algorithm NSGA and NSGA 2, either say non-domination sorting procedure, or the revised procedure that is there in NSGA 2 algorithm.

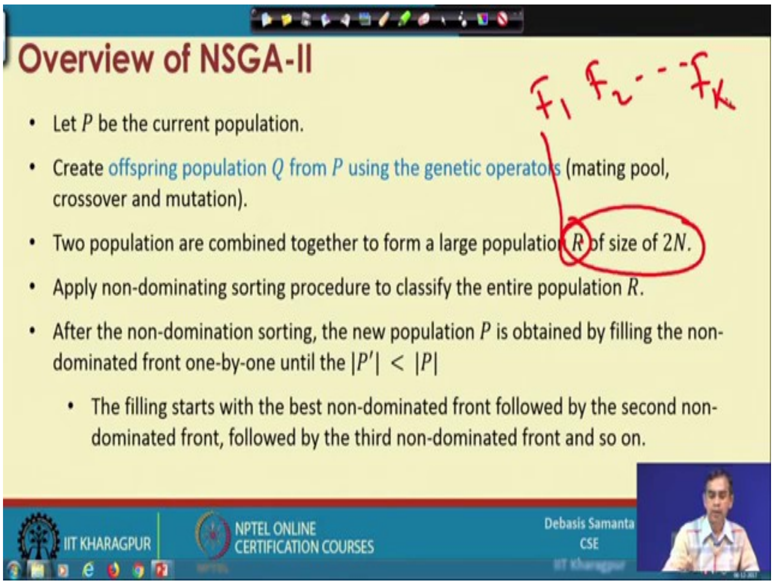
Now, let us see what are the basic step that it is followed and NSGA 2 algorithm, after the calculation of non-dominated sorting front is known to us. So, here is the procedure that I can give in a stepwise manner. So, let P be the current population; that means, the initial so, the current solution or the current generation. Then here the idea is that it basically from the current population P, it generates offspring population Q, right.

And the so, this is basically a reproduction; that means, considering the P, and whatever the method that is known to us in the simple genetic algorithm; that means, a how to create a mating pool the crossover and mutation, if we follow, then from the set P we can deny it another set Q. Basically Q is the next generation like.

So, basically from the current generation P, we shall be able to generate the new solution sets Q based on the reproduction techniques the usual. Now once the P and Q are known what the technique in NSGA 2 is that we combine all the solutions both from P and Q together a solution sets. That mean a P is the initial solution and then Q is the next solutions, then the combining the two solution gives a solution size or a population size let it be R is the size of $2N$. So, R is basically all the solution those are there P and then Q. So, as P size of N, Q is size N. So, the size of $R=2N$.

So, basically, we will developed a large population set of size $2N$.

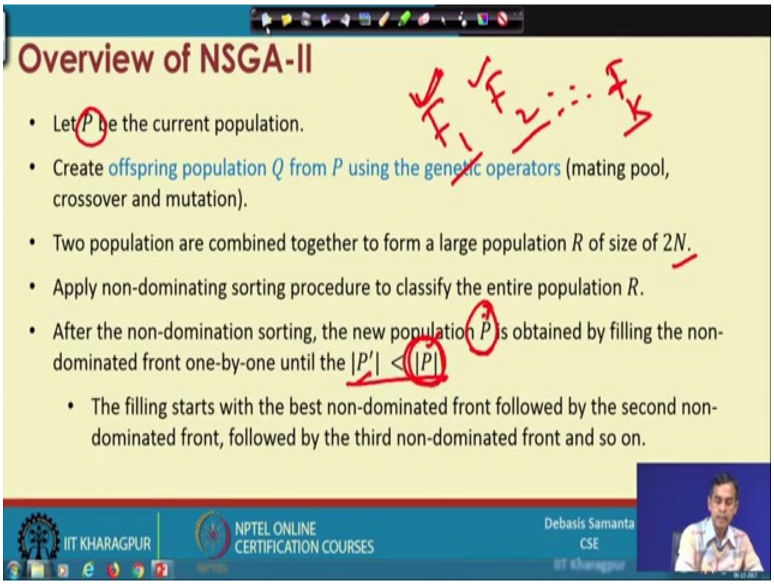
(Refer Slide Time: 11:47)



And then the on this solution set R we have to apply the non-dominated sorting procedure that we have discussed so that all the fronts can be calculated. So, from R we will calculate all fronts. So, $F_1, F_2 \ldots F_k$ number of front say. So, k number of fronts that can be calculated using the non-dominated sorting procedure like.

Now so, after the non-domination sorting is done, the new population P is obtained now. So, this is the current population from this current population the new population P will be obtained by filling the non-dominated front one by one, until the P' is less than P. So, here basically the idea is that so, suppose $F_1, F_2 \, then \ldots F_k$ are the different front.

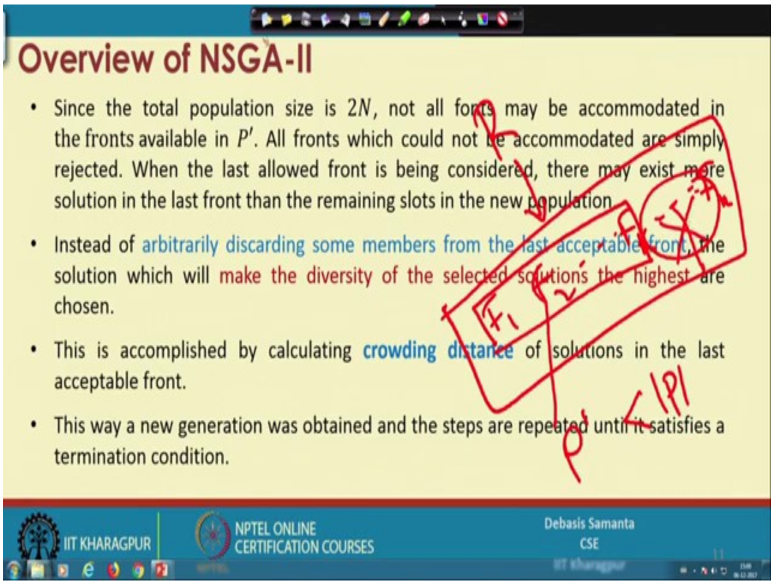
(Refer Slide Time: 12:40)



So, one P so, from the P' so, from the P the next P' has to be obtained ok. So, it is basically we are in the process of getting, the next population set P from the current population P. So, basically, we will select F_1, if we see that size of the next population is less than the population size that is basically N. So, if we select F_1, F_2 and we will go on selecting until this condition is satisfied. And you can say the last front is the one important front, basically, this front if we include into the current population, the lump the limit that is the size of the population will exceeds.

So, that last front is the one important front that needs to be taken into care and the special treatment to be applied. Now we will see exactly what is the special treatment that can be applied to the last front.

(Refer Slide Time: 13:55)



And here is basically the concept of concept of some selection. And that selection is a vital there in NSGA 2 and is basically the in reason behind the success of this algorithm.

Now, here so, so idea it is there. So, since the total population of size $2N$, and we have obtained the many fronts, then all fronts may be may not be accommodated in the fronts available in P'. So, all fronts which can could not be accommodated are simply rejected. Now here again I can say this one say $F_1, F_2, \dots F_k, \dots F_m$. So, these are the all fronts that can be obtained from R so, given R non-dominated sorting procedure will calculate all front.

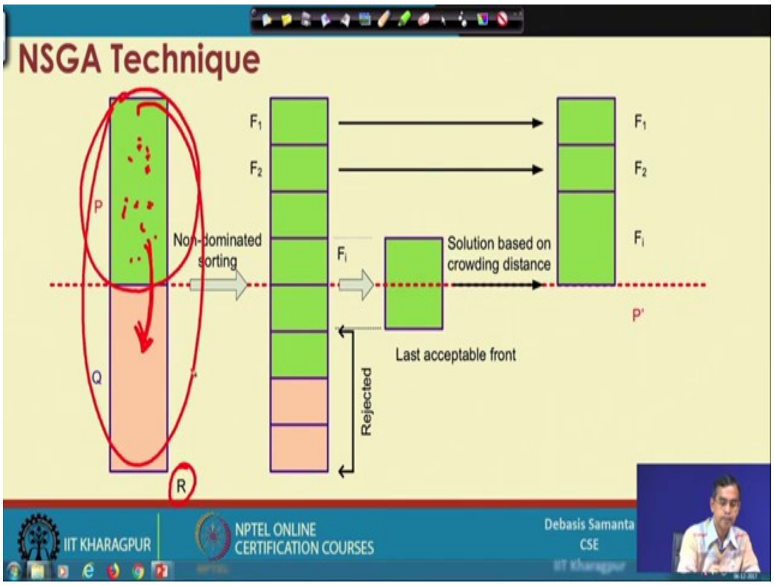
Now, up to which the F_k front can be filled so, to the current population P' like ok. And so, that it is less than the size of P that is the population size. Now, so, the remaining front those are there they can be simply ignored so, they simply reject it. So, this is the procedure it is there, because they are not good for creating the mating pool or next population generation.

Now, here the idea again, instead of arbitrarily discarding some members from the last front, the solutions which will make the diversity of the selected solutions are the highest are should be chosen. So, this is the one important criteria or rationale that so, the last front contains some solutions who is basically then the maximum size of the population, then how to select the solution so, that it exactly the same as the population size. So, basic idea is that the from the last front, we have to select thus that solution which has the very good so far, the population diversity is concept.

Now, in order to select those solutions, which is there in the last front the NSGA 2 procedure, consider one method it is called the crowding distance method. And this crowding distance is basically is idea about that how to select a winner out of the solutions though is those are there in the last front. And so, so this is the basic idea about here, in this technique the crowding distance we will discuss in details in the next slide. So, crowding distance is very important concept there we will discuss these things in the next slides.

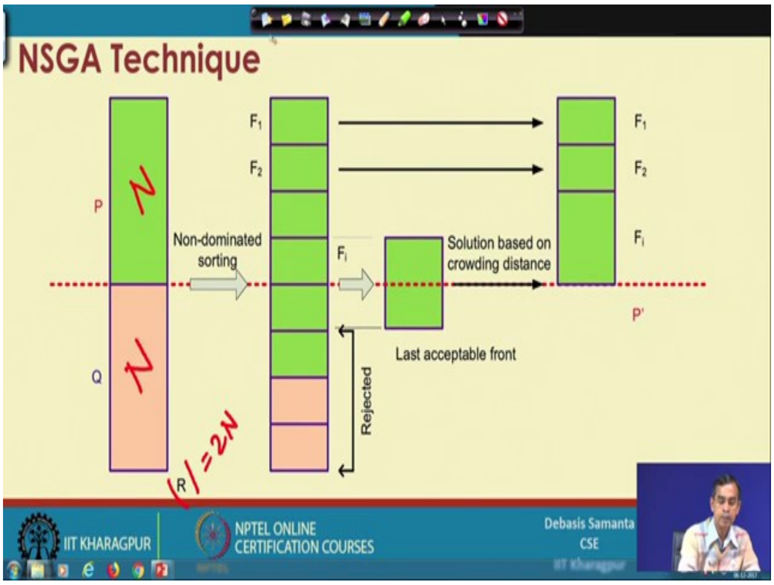
Now, let us first proceed about the other concept it is there ok.

(Refer Slide Time: 16:47)



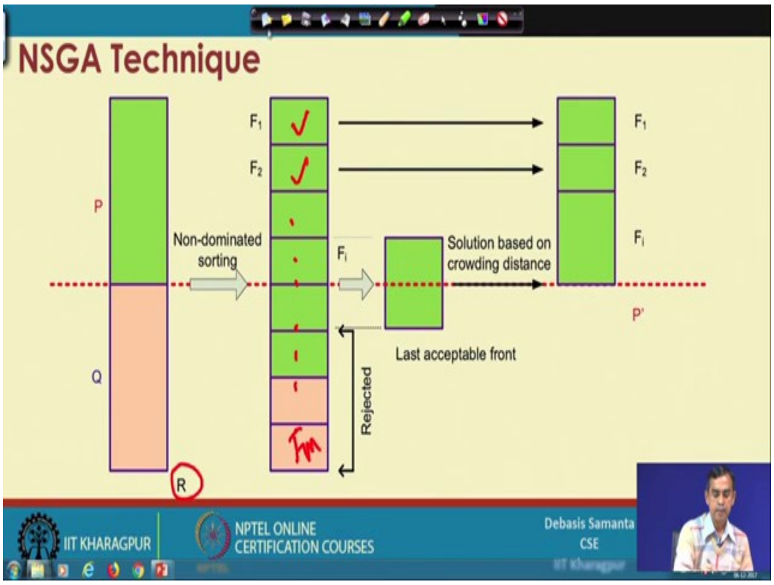
We can summarize the discussion about the basic steps that is there in NSGA 2. So, it is basically NSGA 2 technique, right. Now here we can summarize the thing that we have discussed here. So, these are the current solution in the current generation. So, P and from the P, we produce the Q the next, I mean solutions for the next generation. So, P and use the reproduction method so that from P the Q can be generated. So, all the solutions together it is called R and then the size of $R=2$ to the $2N$, where N is the size of P and N is the size of Q.

(Refer Slide Time: 17:36)



So, these basically the solution sets, that needs to be considered for the NSGA 2 approaches. So, these are first step; that means, the current solution P, and then reproduction after reproduce solution Q are combined together. Now from this current solution P and Q or that are the mark solution R. We have to find the non-dominated front from here.

(Refer Slide Time: 18:10)

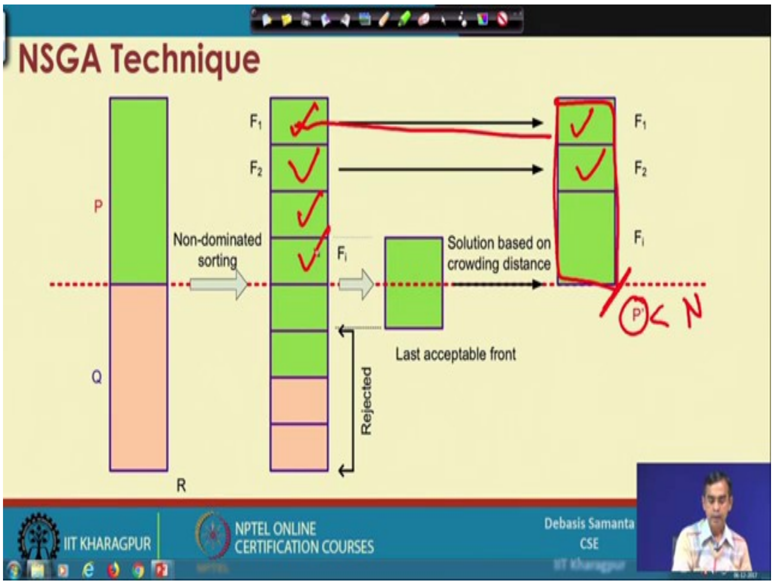


So, so suppose these are the front $F_1, F_2, \ldots say\, F_m$ number of fronts are there. So, they are the separate fronts has been calculated; that means, this is the front 1 first, front

2 second, front 3 third, and so on. So, this is this can be obtained following the non-dominated sorting procedure.

Now, next is basically selection. Now, this selection first is that this F_1 front should be selected if we see that after adding this one, the size of the population will not exceed the quota or limit that is the N .

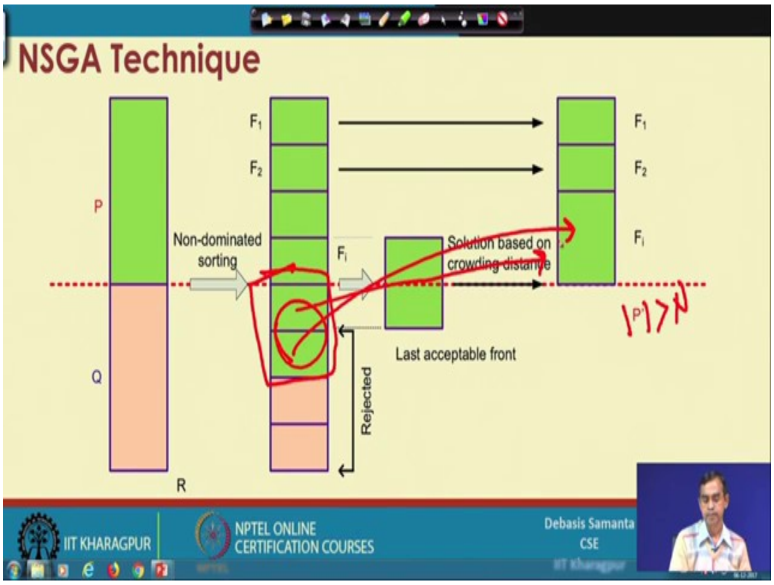
(Refer Slide Time: 18:40)



So, it is basically P' so, we have to go on. So, if F_1 will be added into this P' if we see that size of the P' is less than N . So, the if this is selected similarly F_2 can be added here, if we see that the size of the $P' < N$. So, this way we have to select all this one.

Now, after selecting so many things suppose at this space.

(Refer Slide Time: 19:17)

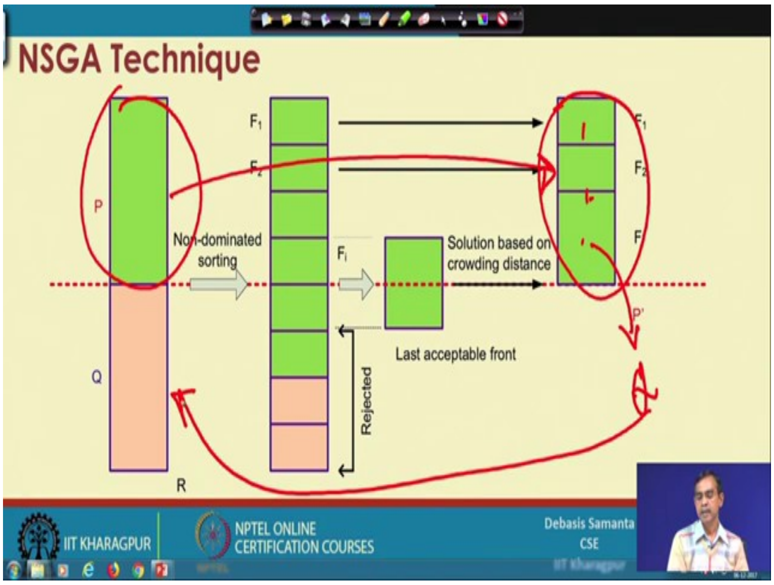


One front is like this, which is the next one to be considered, but if we add this front here, then the size of the $P' > N$. This means, all the solutions which is there cannot be arbitrarily selected into here, right.

So, this is the idea about then this procedure says that then we have to play one tournament among all the solutions which belongs to there, and based on the tournament selection we will select the solutions here to fill the population size $P' = N$.

Now, this tournament selection is basically based on one technique it is called the crowding distance technique. And all the solutions so, from there all the solutions which are the winner to be selected here, and for the rest of the solutions are simply to be rejected.

(Refer Slide Time: 20:23)

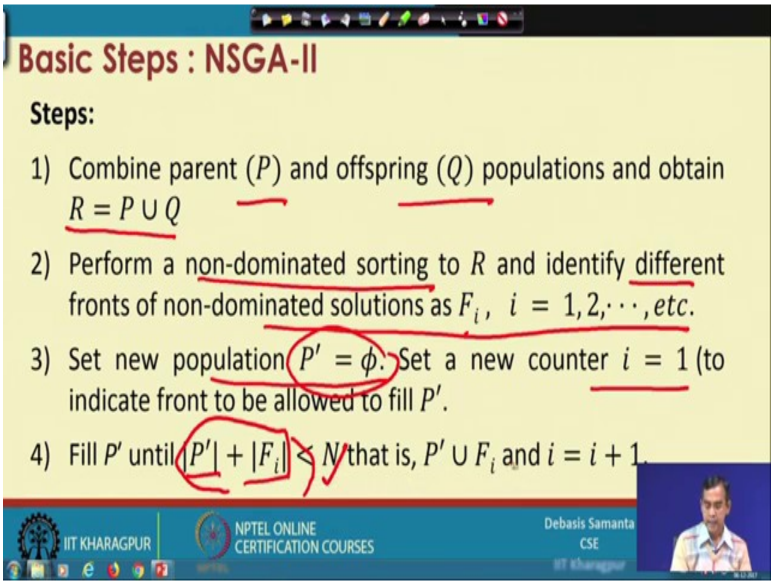


So, this way from the current population P, we will be able to obtain the selected population, and this is basically the mating pool that needs to be considered for the next generation population.

So, from this mating pool we will be able to consider the queue, the next generation population and then same procedure will be repeated again and again until the termination criteria is satisfied. So, this is the idea that the NSGA techniques follows here, but in order to understand the NSGA technique again; that means, here is this procedure is the most important procedure; that means, how the tournament selection based on the crowding distance can be obtained.

So, this is the important thing that is there.

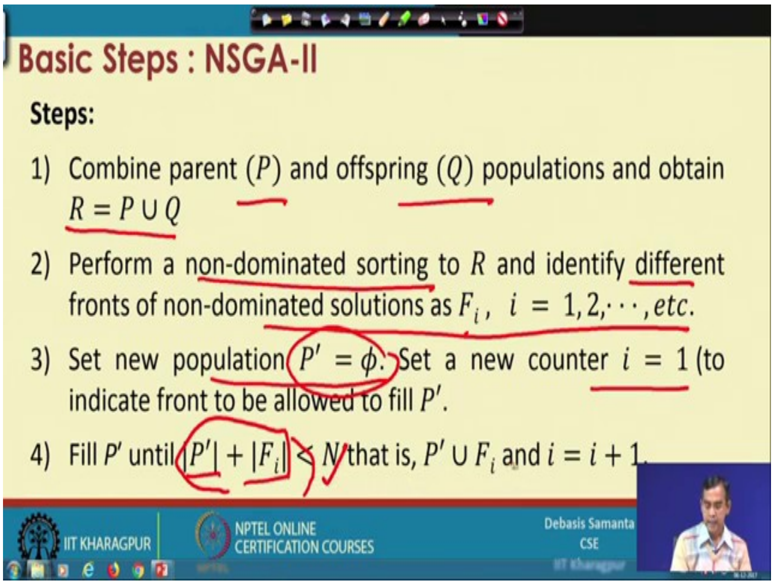
(Refer Slide Time: 21:14)



And so, basic idea again so, whatever the step that we have discussed earlier it can be little bit expressed in a formal way combined parent P , and then offspring solution Q to produce the resultant solution R , perform a non-dominated sorting to R , and identify the different front this is called the non-dominated front F_i like 1 to 2 etc.

Now, set new population P' initial which is empty. And then we have to follow this one, until we have to select the front. Now $P'+F_1$ should be alert. So, that it is ¿N and we stop this selection or adding, when we say that it ¿N . So, for ¿N , we have to consider the method; that means, say selection method that is called the based on the crowding selection. It is called the crowded tournament selection procedure.

(Refer Slide Time: 22:04)



Now, the crowed tournament selection procedure follows one method it is called the crowding sort. For all the solutions which belongs to F_i. And then based on this crowding sort, it will consider $(N - P')$ solution, because these are the solution that needs to be filled up, and that can be based on the crowed tournament selection. And this way the next generation can be obtained and then next offspring generation can be obtained.

So, this is idea about it, now learning the NSGA 2 for this method, that mean how to select the solutions from the last dominated front in order to make the solution size same as the population size. And then this procedure needs the discussion of the crowding sort techniques. So, the crowding sort techniques so, we will discuss about the crowding sort techniques in the ok.

So, we will discuss the crowding sort techniques in the next lectures.

Thank you.

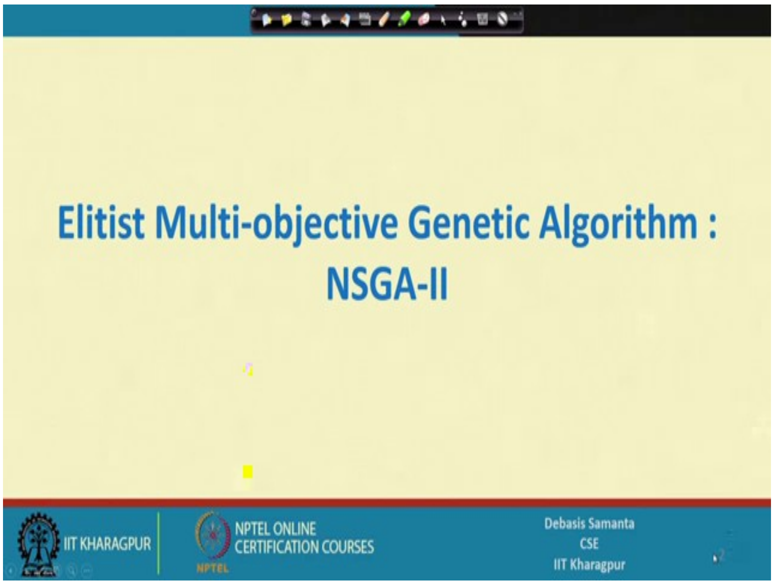
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 33
Pareto-based approach to solve MOOPs (Contd.)

We are discussing NSGA 2 approach, and the NSGA 2 approach followed as some method similar to the NSGA, and it is basically the first method. First step that is they are in both common in a NSGA, NSGA 2 are non-dominated sorting, front calculation based on the non-dominate sorting or a procedure. Now today and then the next procedure that is the here is different is basically the selection for the mating pool. And here in the NSGA we follow the method of assigning fitness values followed by the sharing fitness values.

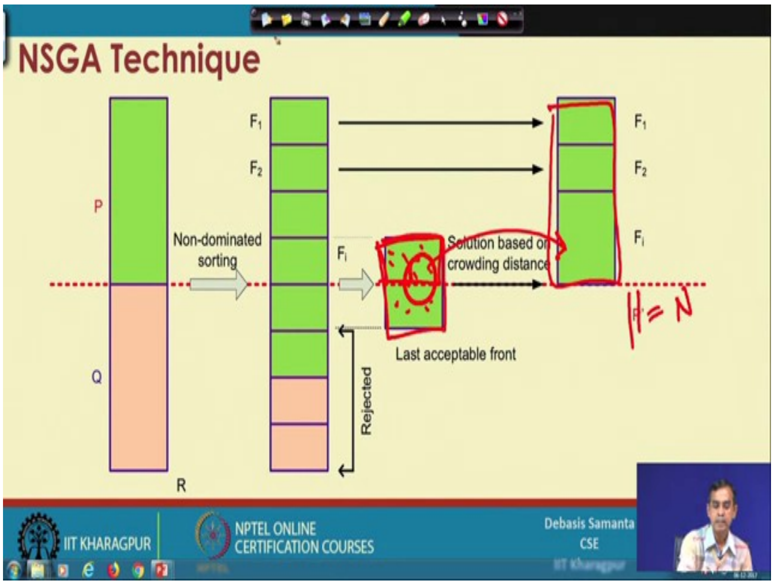
But in case of NSGA2, the method that we follow for the selection is basically called the crowded tournament selection. So, in this lecture we shall learn about this crowded a tournament selection method so, crowded tournament selection method in NSGA 2.

(Refer Slide Time: 01:18)



Now as you know that crowded selection method is basically is crowded tournament selection method is basically required in order to decide from the last front to select the requisite number of solutions to fill this population of size F_1 .

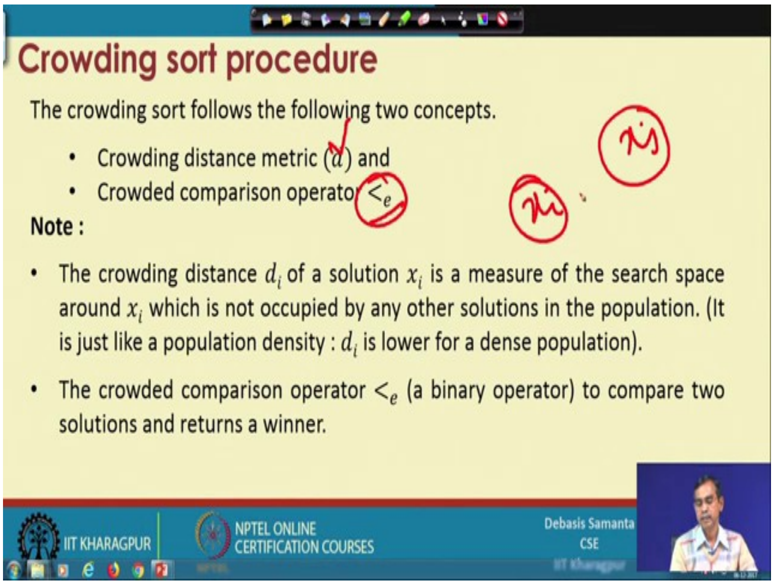
(Refer Slide Time: 01:44)



So, here if this is the last front that needs to be considered to fill this front, but if we include all the solution belong to this front it will exhaust the total capacity, from there we have to calculate only this amount of numbers to be included here. So, that the total size of this solution will be equals to N. So now, how to select the correct solutions are there the most preferred solution to this one, and this method is basically based on the crowded tournament selection method.

Now so, these are crowed tournament selection method will be discussed in these lectures.

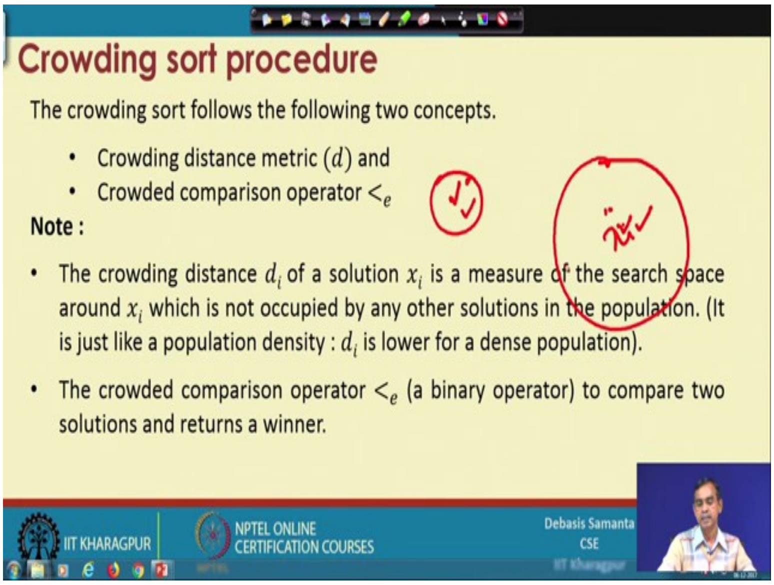
(Refer Slide Time: 02:27)



Now the crowded tournament selection method it is also called crowding sort procedure. It basically considered two concepts. First is that the measurement of crowding distance, and then another is called the crowding comparison operator. So, there is a one metric, if x_i and x_j are given, then how to find the crowding distance. Or rather we can say for every solution x_i , how the crowding distance can be calculated. So, this is denoted as d, and then if x_i , and x_j are the solution, then how we can say that which is the winner that is based on the crowding distance measure that you have because, x_i has it is own crowding distance x_j is on crowding distance.

Then selection is based on an operator, that is called the crowding distance operator that can select the winner. That means, is compare based on this operator this one. Now crowding distance d_i if we denote of a solution x_i is in fact, is a measure of the search space around x_i which is not occupied by any other solution in the population. So, physical meaning of this crowding distances like this one.

(Refer Slide Time: 03:53)



If x_i this is the one solution here, and then how many space it is there, by which the next the neighbor is there. So, this is the crowding distance. Now if this is the solution x_i, and this is the crowding distance for the next neighbor it is there, then we can say that crowding distance of this solution is more than the crowding distance of this solution.

Now, another physical meaning is that if the crowding distance is very large. This means that this is a one solution, which is there in a less populated on region. On the other hand, if the crowding distance is small, then we can say that this solution is belongs to one solution, which is heavily the populated. So, is basically crowding distance says, that whether this solution is in a crowded region, or this solution is in a crowded region or both the solutions in a crowded region, then which solution are in a heavily crowded region than the others.

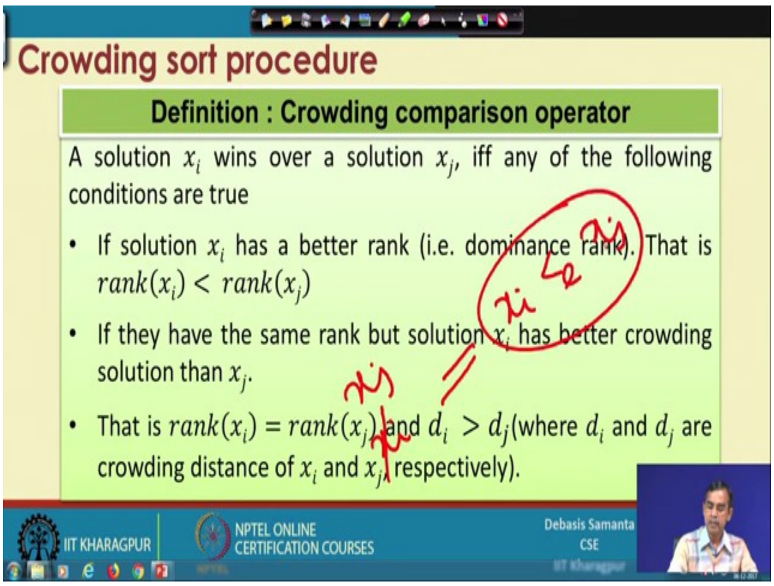
So, this is the meaning of this crowding distance concept it is there. And and then crowding operator this one is basically to compare the two solution, so far, their crowding distance is concerned. So, these are the two things are the here and we will discuss about these two things here. Now let us first define the crowding comparison operator that just now we have discussed by which the two solutions can be compared and based on this comparison we can select the best solution here.

Now, let us consider x_i and x_j are the two solutions, and they are the crowd crowding distance is known to us also. Now so, crowding comparison operator is basically is a operator which is defined here how this operator works for us. So, here the operator is defined like this, if solution there are two conditions actually, if solution x_i has a better rank; that is, rank x_i is better than rank x_j ; that means, x_i in higher rank, I mean is a better rank than x_j .

Now, rank actually you can remember I told once that all the solutions which are the first part, they can be considered the higher rank, and then next solutions which are in the next rank is the next rank and so on. So, x_i is the first front and x_j is the next front then we can say that this one. So, here this is the one condition that is to be satisfied then you can say that x_i is the winner than the x_j . On the other hand, there is another condition if they had the same rank, but solution x_i has better crowding solution then x_j ; that means, it has better crowding distance then x_i then x_i can be considered the winner than the x_j .

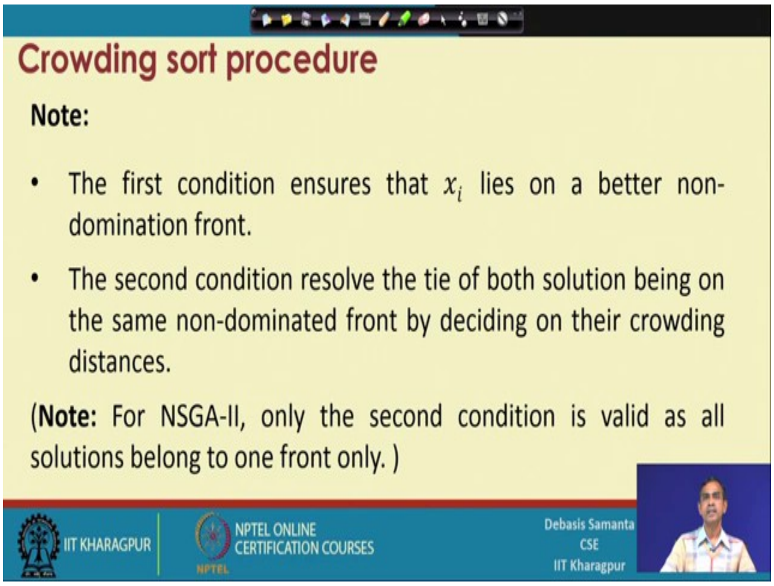
So, the two conditions are to be satisfied, and based on these things it will basically select the selects the winner. So, the conditions again rank x_i if it is rank x_j and d_i is greater than d_j , then x_i is the say operator. So, this is the idea about the crowding operator.

(Refer Slide Time: 07:19)



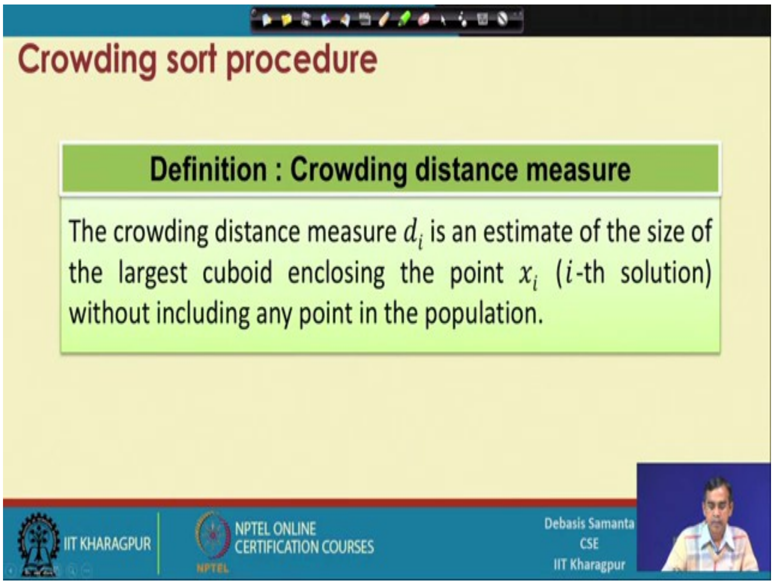
So, this means if x_i, and this is the crowding operator x_j is basically checked out of the x_i and x_j; which has to be returned. So, either x_i or x_j so, based on this condition, this operator is defined here. So, this is the concept that is there so far, the crowding sort is concerned. Now, let us see how this concept is there.

(Refer Slide Time: 07:45)



In our case, if we see in the in our case; that means, we are to consider all the solutions belongs to a particular front that is the last front. So, therefore, rank is not required the first condition need not to be satisfied. Because all the solutions belong to this front have the same rank. So, that is why the first condition is not need to be checked there. Only the second condition needs to be checked. So, second condition resolved the tie, when basically if both solution belong to the same front, but the solution that tie can be resolved by means of calculation of their crowding distance. Or in other words, in NSGA 2 only the second condition is valid as all solutions belong to only one front.

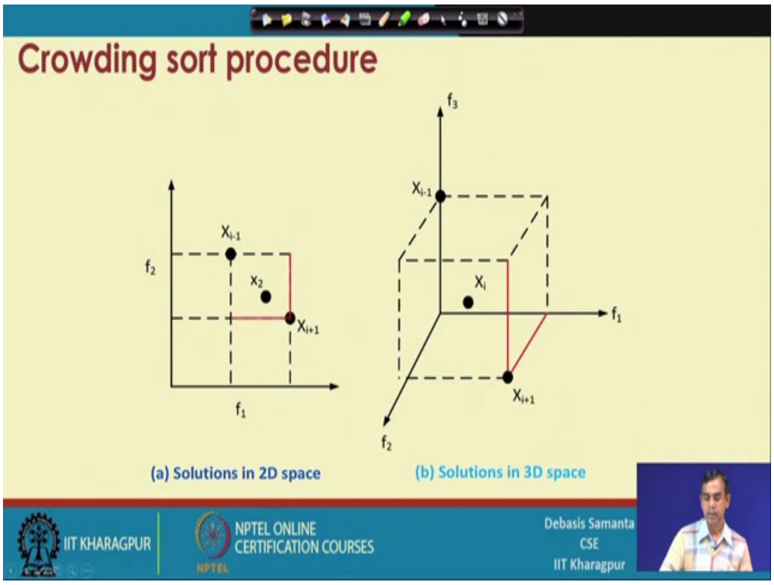
(Refer Slide Time: 08:42)



Now, let us see how the crowding distance measure can be carried out, we have already told you that crowding distance measure is basically the population density surrounding a solution. But how we can measure this population density? NSGA 2 follow a cleaver approach to do these things. So, according to this NSGA 2 crowding measure distance d_i for a solution x_i ; that means, is the i-th solution is an estimate of the size of the largest cuboid enclosing the point x_i , without including any other point in the population.

So, this is the definition actually. So, that is a largest cuboid enclosing the point x_i , this is important; that means, if x_i is given to us, and if we are able to find the largest cuboid which surrounding the x_i . So that there is no any other point in that cube, then that cube will give a measure to the crowding distance. So, this is the idea about the definition of crowding distance and as it is there, I can illustrate the same concept with an example here, let us follow.

(Refer Slide Time: 09:56)



So, first consider this example, and it is a case of two dimensional; that means, 2 objectives f_1 and f_2, and for solution say x_2 we want to find the crowding distance. Now surrounding this two point, the two nearest point with respect to x_2 is this one say x_{i-1} and x_{i+1}. These are the two nearest point, with respect to the solution point x_2. Now then with x the two nearest points; that is, x_2 with respect to this we can find one what is called the region it is there.

So, this region is basically either crowding region here; that means, surrounding this x_2 these are the basically area by which no other points are enclosed. So, the x_2 is the crowding measure here, now the x_2 measures is basically the crowding measure is ok, we can take the calculation of the square of course, area, but this this energy to propose the major that this plus this is the measure of this crowding distance it is also alternatively, because if these two measures basically breadth and width measure in this case.

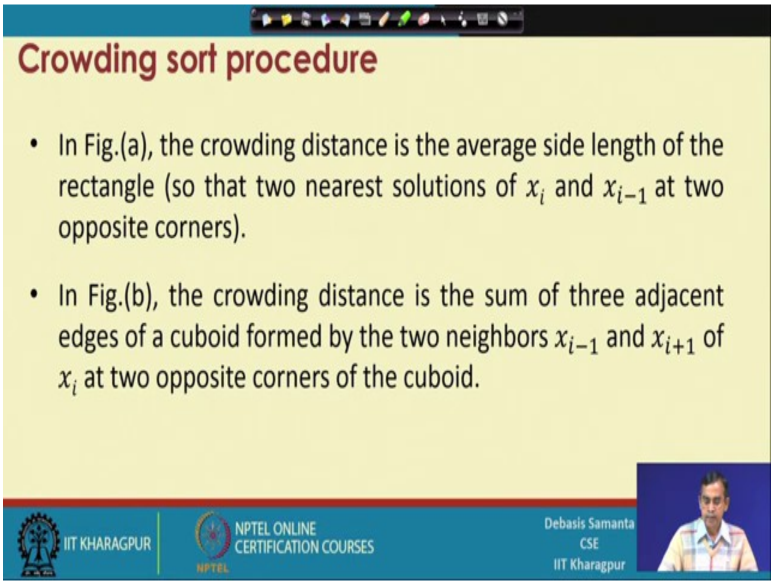
Now so, this way if we know x_2, and these are the two solutions are there we will be able to find these two distance and therefore, the crowding distance can be measured. So, this is the one example where the crowding distance how it can be measured here, and another example. So, this is another example here, three objectives are there. So, is a multi-objective optimization problem with three objective function f_1, f_2 and f_3 ? And we are interested to find the crowding distance for the solution x_i. And suppose, x_{i+1} and x_{i-1} are the two solutions, which are the nearest to x_i , they are the near most two solutions.

Now, if we can find the two solutions, which are near most to this one, then in 3-dimension unlike this 2-dimension we can find a cuboid. So, this is a cuboid and this one and then the crowding distance of this thing is basically this is the total area of the cuboid. But instead of calculation area of the cuboid, it will take the calculation of this and this are the measure of the size of the cuboid. So, this will give the measure of the cuboid, and then this can be given alternative measure or is basically the measure of the crowding distance.

So, we have learned about how the crowding distance for any solution in a 2-dimensional space can be calculated for any solution in 3-dimensional space can be calculated. Now extending the same idea, we will be able to calculate the crowding distance for any solutions, but in a N-dimensional space. Now we will give an idea or the formula that that initiative others proposed it, how the crowding distance can be calculated will be discussed.

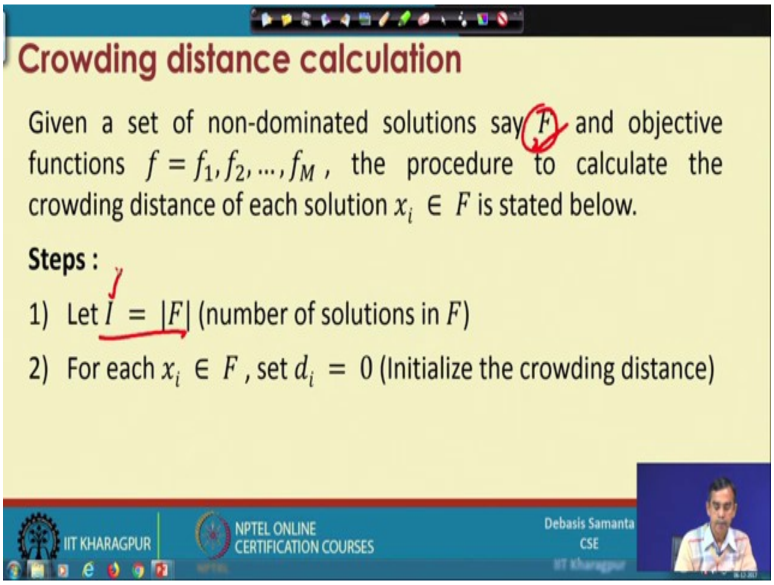
So, anyway so, crowding distance can be calculated, if a solution is given knowing the other solutions in the near dignity of the solutions.

(Refer Slide Time: 13:27)



Then the crowding so, is basically the crowding distance and then once the. So, the idea is that for all the solutions, which are belongs to a to the last non-dominated front for all the solution belong to the last dominated front. We have to calculate the crowding distance for all. So, this is the step that is required here.

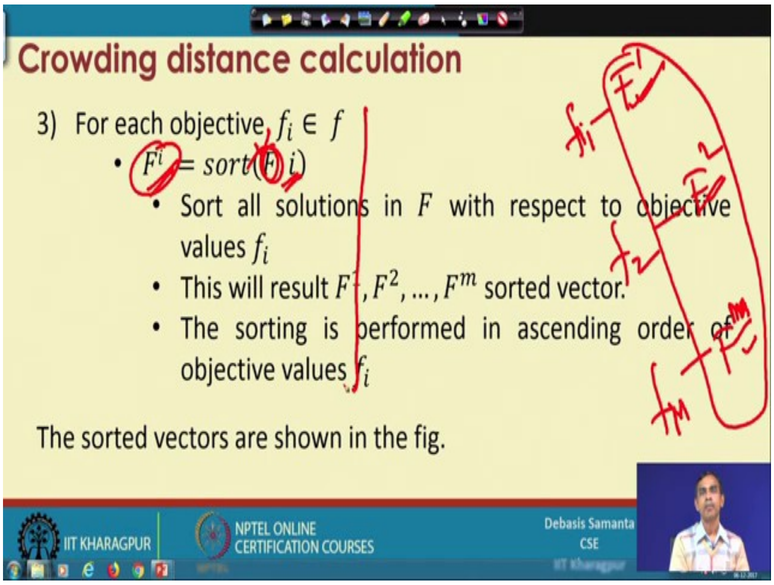
(Refer Slide Time: 13:57)



Now, crowding distance calculation can be carried out in a little bit mathematical way, that just I want to discuss it here. See suppose, given a set of a non-dominated front the last non-dominant front that is here let this be F. And they are objective F objective function for each solution they are denoted as $f_1, f_2 \cdots f_m$. So, for each solution we have this objective vector F with in terms of m objective functions. And let the size of the F be this one; that means, the number of solutions which belongs to the non-dominated front this be the I.

So, here basically the procedure is that for each $x_i \in F$ set $d_i = 0$, initially the crowding distance of all the solutions is 0. And then we will calculate for each solution $x_i \in F$ what is the crowding distance, we have to calculate it initialize 0, and then finally, you have to calculate the crowding distance for all solutions. Now here is the procedure for each objective $f_i \in f$, basically we will first sort all the set f, but with respect to the i-th objective vector and this be the sorted f.

(Refer Slide Time: 15:09)

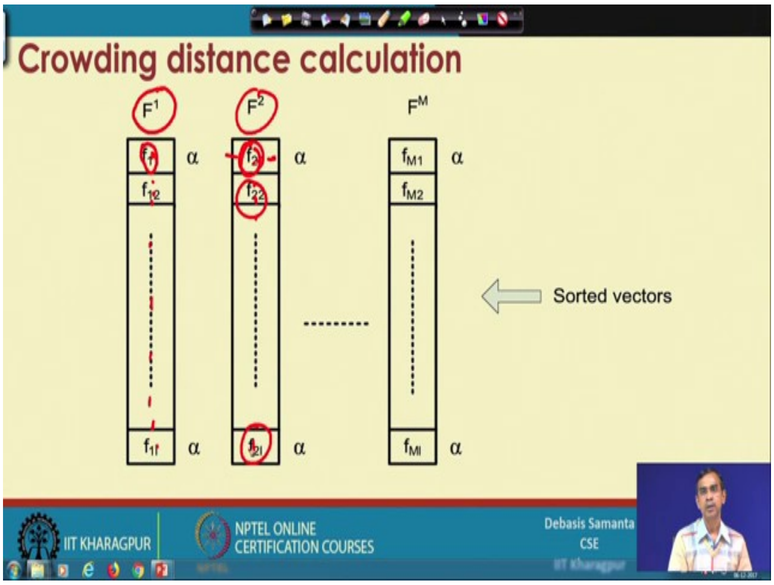


So, it is that idea it is like this.

So, first with respect to f, f_1 objective function we are to sort all the solutions which belongs to F and then the shortage solution will be termed as F^1. Similarly, with respect to f_2 if we sort all the solution belongs to F and it will give F^2. So, this way if with respect to f_M, we can get the certain percent of sorry sort F^m. Now here we can see so these are the sets is a sorted order, but is a sorted order with respect to one component. This one sorted with respect to F^1, this one is sorted is F^2, and this one is sorted F^2, F^m and so on.

So, here basically a sorting techniques are to be followed by which all the solutions belongs to the set belongs to the set F can be sorted in terms of one objective function at a time. So, these are called the sorted vectors. So, pictorially all the sorted vectors can be shown like this, if you can see this figure here.

(Refer Slide Time: 16:46)



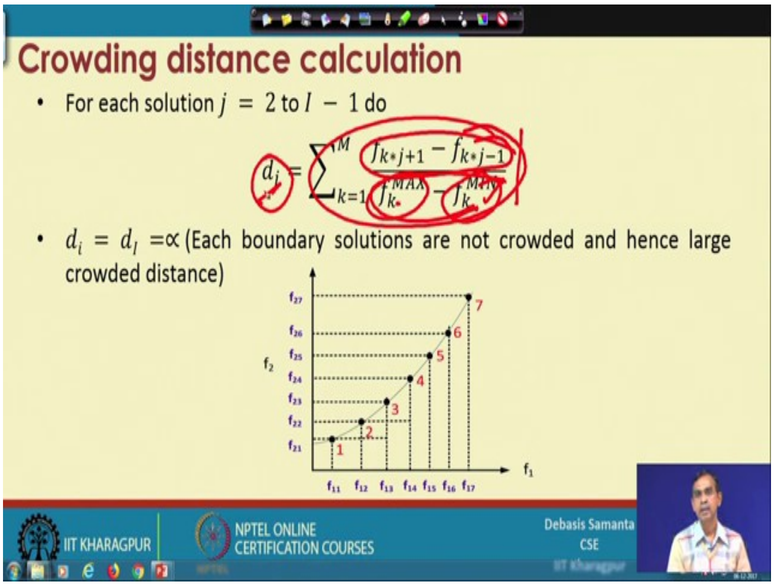
So, these are the sorted vectors, you can see F^1 sorted vector with respect to the objective function f_1. So, here basically all the solutions, they are sorted in ascending order, but with respect to f_1; that means, the solutions has the lowest value. So, for the f_1 is concern this is the next highest value, and it is the highest value of F^1 is concerned.

Now, again and this is the solution F^2 is respect to the second objective function; that means, with respective second objective function, all the solutions are there which has the lowest value of F^2, then this solution which has a next higher value and so on and this basically the solution which has the highest value; so, for the objective F^2 is concerned. So, this way the F^1 the sorted vector F^2 and F^M, and here with respect to F^M can be often.

Now, in this discussion we assume on concept is that here all the objective function are to be minimized. So now, if it is not minimized the other if it is to a maximize, then we will follow the descending order. So, if it is minimized, if it is maximized, then it is ascending order it is descending order. So, this procedure depending on, we can consider that all objective functions are to be minimized one. So, if it is a minimize, then all these are the ascending order of their objective function.

So, this way the sorted vector can be obtained. Once the sorted vector is obtained, then we shall be able to calculate the crowding distance between or crowding distance of each solutions easily. So, this method that is proposed in the NSGA 2 is like this.

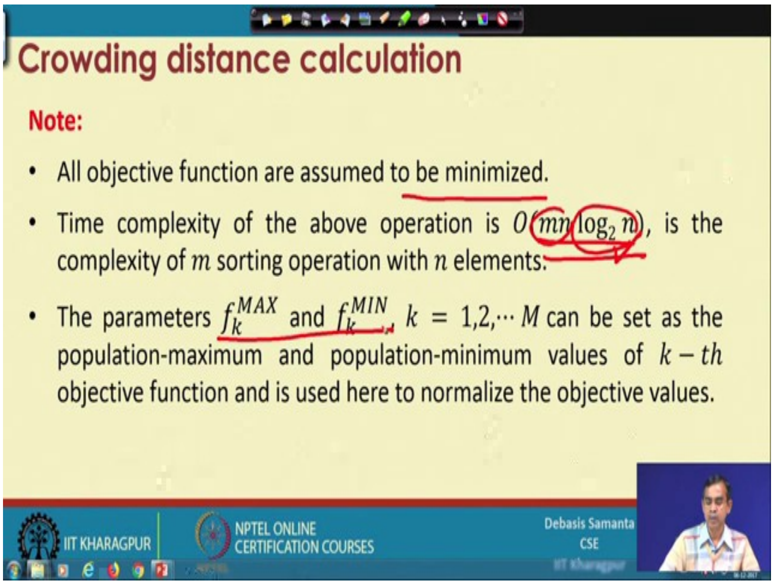
(Refer Slide Time: 18:35)



So, the crowding distance d_j for any j-th solution can be obtained like this one using this formula. This formula can be checked yourself, and you can find it here the f_k^{MAX} and f_k^{MIN} are the two values. It means with respect to the k-th objective function or is the lower bound and is the upper bound. And this value is used to normalize all the solution, because is a normalization is require so that all values of the d_j will be in a same range. So, this is for the normalization, and this formula can be verified yourself. And another thing is that, the first solution and the last solution this is basically di the last solution, because it is a boundary solution, they have crowding distance is infinite. So, this is the condition.

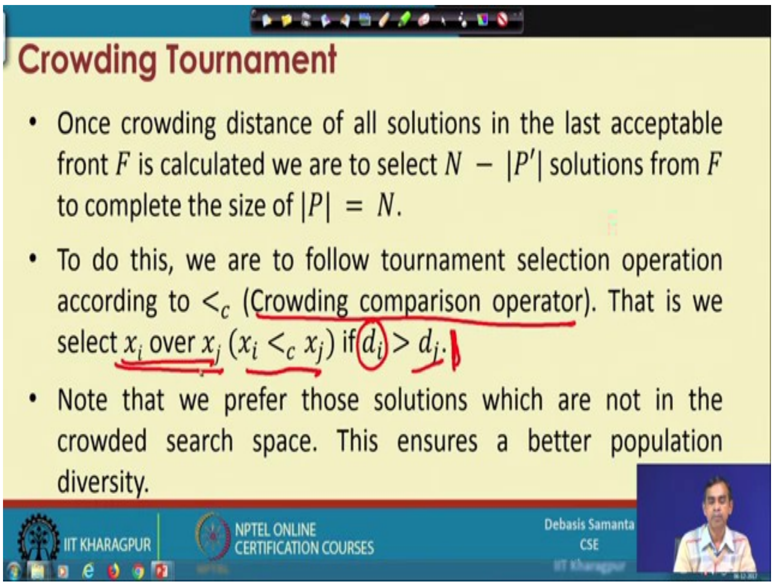
So, this way, we shall be able to calculate the crowding distance of all the solutions there. Now once the crowding distance of all solutions are calculated, then the then we can play the crowding sort procedure here ok.

(Refer Slide Time: 19:38)



So, here we have already you have mentioned that all objective functions are to be minimized in our previous discussion. And so, far the complexity of the crowding distance calculation is concerned, because it is a sorting method. So, it is a sorting complexity $O(mn\log_2 n)$ where n is the size of the population. So, is the complexity is this one. And f_k^{MAX} and f_k^{MIN} are the two limiting values, and then we divide the this limiting a difference between the limiting values to normalize objective values ok.

(Refer Slide Time: 20:20)

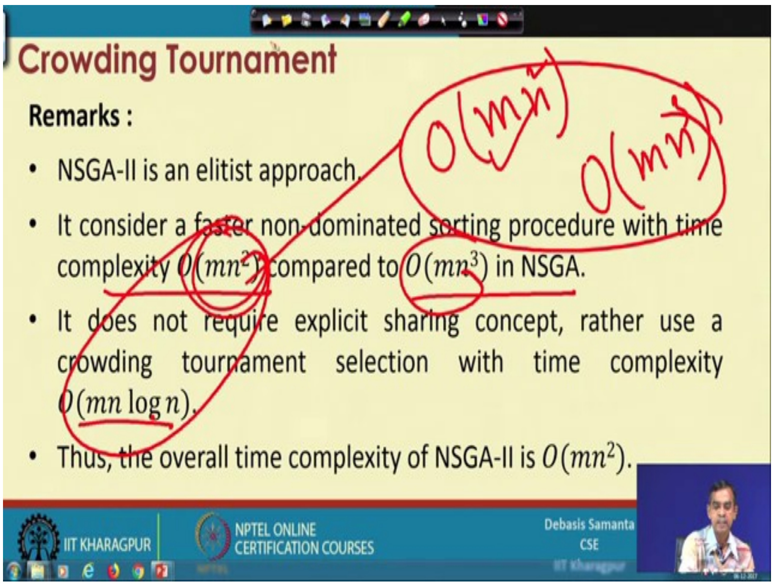


So, crowding distant calculation once it is done, then we will be able to play the crowding tournament selection game.

So, crowding distance selection game is basically it is like this we can follow the crowding comparison operator. Now here we ok. So, crowding operator can be applied to the two solutions x_i and x_j ; that means, we have to see this one. Now here $¿_c$ crowding operator x_i , and x_j . x_i will be termed as a winner if we see that, we are crowding distant di is greater than d_j . So, in this case this is a crowding operator. Here and you can see again that all the solutions are get on the same rank, that is a you do not have to bother about rank.

So, here basically so, for the crowding distance based tournament is concerned. We prefer those solutions, which are not in the crowded search space. This ensures a better population diversity. So, basically $d_i > d_j$; that means, the solution x_i is in not in a crowded region. Now so, this basically in to ensure the population diversity. It is same concept that is where in NSGA, but NSGA follow. Then is count here instead of needs count it basically consider crowding distance. So, this is the difference between NSGA and NSGA 2 is here.

(Refer Slide Time: 21:48)



Now, NSGA 2 is an elitist approach. Why we said so? This is because, if you see the non-dominated front, it basically when we match the two solutions the previous generation and the next solution, and from there if we find the non-domination front, then it basically selects the all elite solutions first. So, that is why it is called the elitist approach. Because, first front, second front, third front and the elitist fronts are selected first and for the last front it is the lowest or worst elitist front from there we have to select using the crowding tournament selection.

So, here this is why the concept it is there. And so, far the procedure the time complicity is concerned. The total time complexity here is $O(mn^2)$ compare to $O(mn^3)$ there in NSGA concept. And it does not require any explicit sharing concept that is therein can of in case of NSGA method. Rather, it uses a crowding tournament selections with and complexity $O(mn \log n)$. Now so, the two-time complexity $O(mn \log n)$ for the crowding tournaments and for this $O(mn^2)$ for the non-dominated sorting procedure, putting together putting these 2 operations basically complexity the $O(mn^2)$ only because this is the higher bound than this one. So, it is basically taken this one.

On the other hand, in case of non-NSGA it is $O(mn^3)$. And, obviously, So, it is a $O(mn^3)$. And out of these two complexity, this is the time completely with the lower effort than this one. So, that is why NSGA is the first method. So, it is a first method, and it is an elitist method. So, this is the idea about the here, and accuracy it is observe that, this algorithm gives better result compared to both the compared to any pareto based approach that we know so far; that means, moga approach NPGA or NSGA it gives better result and with the fewer competition compare to NSGA of course. And obviously, if you consider time complexity, then it needs more time compared to moga and NPGA however, but the complicity is better.

So, these are the different pareto based approach we have learnt. And what I want to say in the summary is that, out of the different approaches to solve multi-objective optimization problem, non-pareto based approach needs a prior knowledge. Whereas, pareto based approach does not require any prior knowledge. So, this is the one advantage that the pareto based approach is there. And another difference between the non-pareto and pareto that, non-pareto gives only one solution. But all the pareto based approach gives pareto optimal solution; that means, trade-off solutions. And then are from the trade-off solutions, we have to decide one solution, and that solution require some posterior knowledge. That mean it is depends on your decision that out of these solutions which solution can be consider.

But in case of pareto based solutions or pareto optimal solutions, we can select any solutions out of a large set of solutions that can be said that can satisfy your requirement. So, this is the difference between the pareto based and non-pareto based approach. And as we told you the non-pareto based approach it is applicable, if we see that only few are trade-off solutions are to be considered or it is there in the problem solving.

On the other hand, we should apply pareto based if we see that there is a large number of solutions are possible which are equally towards the optimum solution. So, in that case we follow pareto based approach. Out of this non pareto and pareto based approach in fact, people prefers pareto based approach because of it is accuracy and then performance first. So, this is the technique that we have learned about multi-objective optimization problem solving using genetic algorithm.

So, our next topic in the next field will discuss about a neural network concept to solve some computing problem in different application.

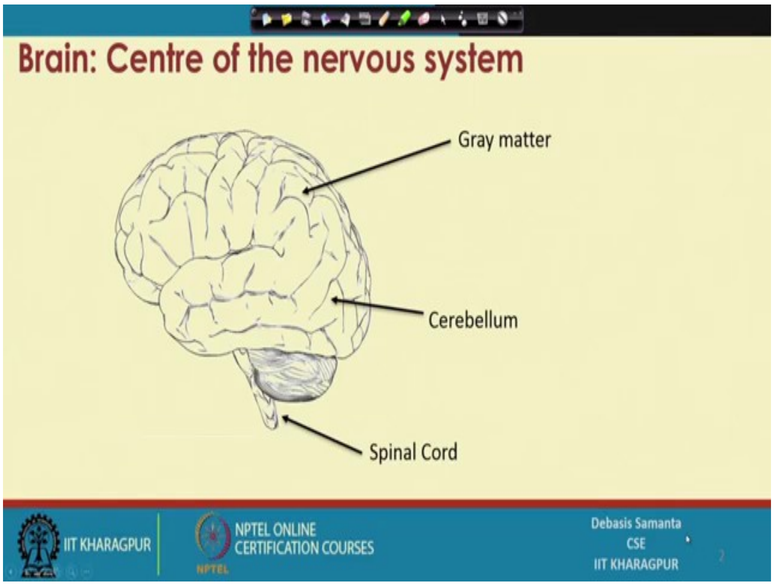
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 34
Introduction to Artificial Neural Network

In soft computing, one another important paradigm of computing is artificial neural network. So, in this lecture we will introduce the concept of artificial neural network and it is uses to solve many problems in different applications.

(Refer Slide Time: 00:44)

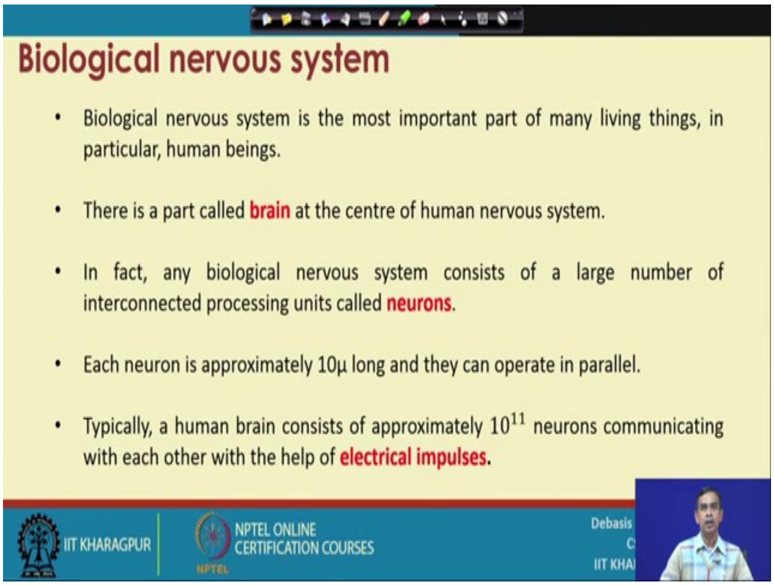


So, we know the human is the best creature in this universe and the main things, that is intrinsic in the human is basically it is brain. Brain is also called central nervous system due to this very unique characteristics of the brain, human can do many thing human can remember, human can reason out, human can prove theorem, human can solve many problems human can see the world recognize those things and many more. So, behind all these the all performance compared to the other living things in this world the human play the brain plays an important role.

Now, as the brain it is also central nervous system ok; biologically it looks like a gray matter. So, that is it why sometimes in medical science brain is called the gray matters. Now, in this gray matters there are a lot of other brain cells are there and any things from any part of the brain is basically controlled by this central nervous system. So, this is this how this brain is also called the head office of our body.

Now, today we will see exactly how this brain is composed of and how this brain works and then how the same thing can be mimicked to solve our problem in an artificial manner so, the artificial neural network.

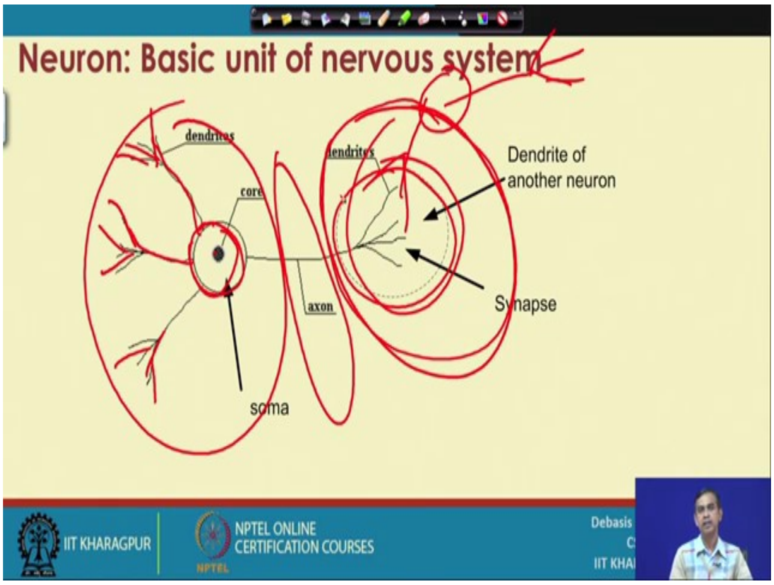
(Refer Slide Time: 02:41)



Now, so in the brain in fact, there is a large collection of brain cells as I told you this brain cells is, basically the atomic level the processing units and more precisely this atomic units is called neuron. Each neuron is approximately in micron in length and these are the unique neurons which basically are the fundamental things of any what is called a sense processing.

Typically, within a human brain there is around 10^{11} number of neurons. And these neurons are basically stay there in a connected manner or you can say in a network manner and in this network all these neurons are the units which basically carry certain pulses. This pulses basically same as the electrical pulses. So, it is also in many ways similar the way how the current flows from one source to another destination. So, these neurons are the cells which basically propagate the electrical pulses from any part of our body to the central nervous system and vice versa. So, these neurons are the important things and we will see exactly how a neuron looked like.

(Refer Slide Time: 04:18)



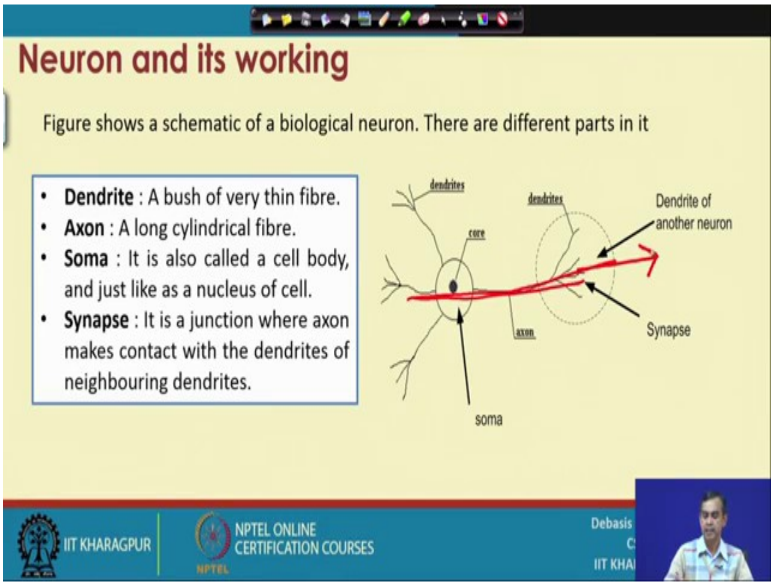
So, in this slides we see one neuron and if we see these slides then you can understand it has three different parts. So, this is the first part, this is the second, and this is the third part. Now, this part is called the head of the neuron. Now so, in this part one is a elongated or is a soiled portion is called a cell body of the neuron and it is called the soma and in the soma there is a core this core is not exactly the nucleus as it is there in the body cell. And, now in the soma there will be hairy like connection these are called dendrite. Dendrite is very small thin hair like organs parts.

And, then the next part it is basically end or tail of the neuron. It is called the it is called the synapses. So, basically the synapses is one part where it basically meets with other dendrites of other neuron. So, it is basically a junction point of meeting other neuron. So, other neuron. So, this is a junction point. So, there is a synapse is also called junction point. Now, between these soma and synapse; there is a connectivity this connectivity is called the action. So, this way the neurons are constructed.

Now, this neuron just like a body cell it is also a cell. It is a living cell and the important difference between the other body cell. Then this nerve cell is that the other body cell can go cell division whereas, the neuron cannot go cell division this means that at the time of birth a person having number of neurons can never be increased. And also if some neurons are damaged or destroyed it cannot be reproduced unlike the body cell, if there is a cart or wound it will be healed and then some new cells will grow to fill the wound or heal. So, this is the difference between these cells and functionally there are many differences between these neurons and the simple body cells ok.

So, we will learn about the neuron. So, neuron is look like this.

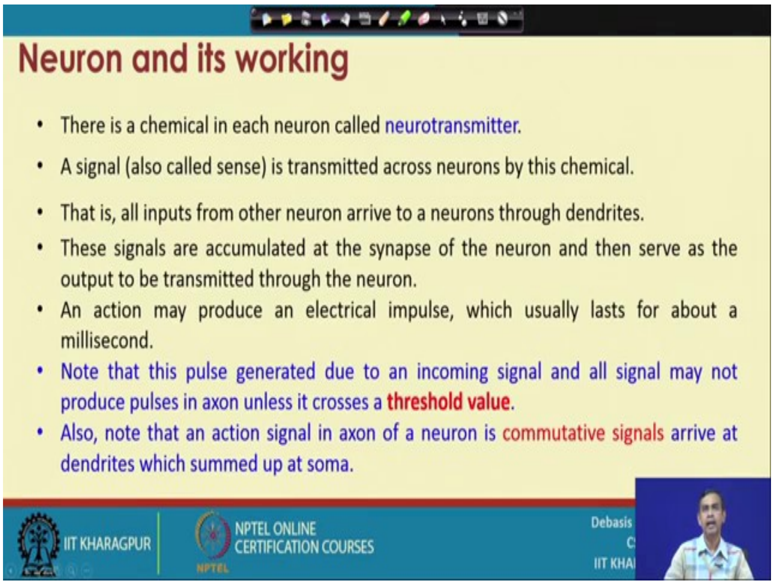
(Refer Slide Time: 07:11)



And now let us see how this neuron is basically work there now and this is a very very schematic of a biological neuron and the different parts that does know we have discussed about. So, different part means the dendrite, the axon, soma, and synapse and here the signal, signal will flow from dendrite to axon; that means, from one neuron to the next neuron. So, this way the signal can propagate it in a one direction. So, if; so, there is a basically connection from every points in our body to the brain and that is the network is there and for building such a network the basic unit is basically this neuron ok.

So, this is the neuron there now here one question that arises is that; how the signals flow from one cell to another cell.

(Refer Slide Time: 08:11)



Now, in every neuron there is one sort of fluid is there. Those fluids are called neurotransmitter. That means, the body of a neuron is filled with this liquid it is a neurotransmitter now a signal whenever it is created this causes some what is called a different level of concentration. So, far this liquid neurotransmitter is concerned for example, if a mosquito bites then the at the point where the mosquito bites at that point a signal is created, the signal is basically is it created means it basically creates a different level of what is called an neurotransmission concentration.

Now, this neurotransmitter is basically is a solution sort of thing we can in a simple manner we can say is a some concentration of some cation like any sodium calcium magnesium all these things are there. So, these are the basically is ion concentration. So, whenever a signal or some event occurs then there is chain in this concentration level of these ions as a result some voltage will be developed and due to this voltage this signal will propagate from one neuron to next neuron.

So, this is nothing, but an in just like is an electrical impulse and this electrical impulse, whenever it is created in a neuron lasts only for few seconds it is not few seconds rather it is for a few milliseconds; that means, whenever that ion concentration difference occurs it will persist only for a few milliseconds after that again concentration will be balanced and there will be no signal or no pulses and so, so this way the signals are created and once the signals are created. Signals will be propagated from one neuron to another neuron.

Now, in this context one thing we should note that all signals cannot be propagated from one neuron to another neuron. A signals which have certain, what is called the strength more than a threshold value only can be transmitted from one neuron to another neuron. If the signal strength is less than this threshold value, the signal will not be transmitted from one signal to another signal and another from one neuron to another neuron and another important thing is that to a neuron the signal can arrives through the different dendrites and.

So, many signals whenever coming from the different neurons to a particular neuron are summed up summed up at the soma and then summed up signal is basically propagated via axon through the synapse to other neuron. So, these are the things that happens in our biological neurons.

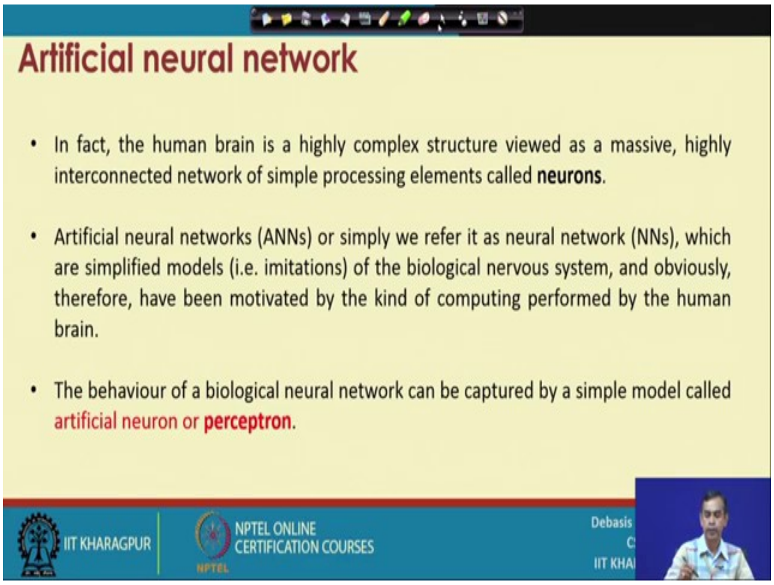
(Refer Slide Time: 11:30)

And this idea is enough to understand how these things can be considered to solve many problems.

Now, see these pictures here how the signals can be. So, here basically one, here basically some event occurs. So, these basically produce some, what is called electrical pulses will be flow there come here and then go there this way it will flow and the signal which is produced here right. Because I told you once point here, but in this point the number of neurons are n fact, located.

So, so the point where the neurons are located they will receive this pulse and then pass through this what is called a neuron and then summed up here and when this signal strength is greater than a threshold value will be passed through these synapse and then from there it will go to the other neuron. So, this way the signal propagation takes place in our neuron.

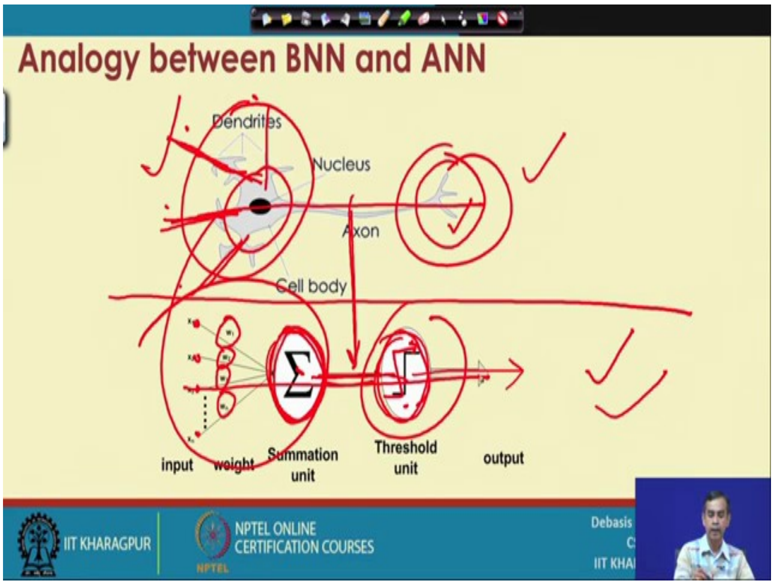
(Refer Slide Time: 12:36)



Now so, this is the idea that is what is called the biology biological neuron. In fact, the human brain is basically the very complex structures and it can be viewed as a massive, highly interconnected network of these neurons. So, gray matter that we just have now learn about it is basically nothing, but a collection of neurons, as I told you it is around 10^{11} neurons. The people who are having more neurons they have the more processing or computing capabilities thinking capability they are great scientists like Albert Einstein.

Now these artificial neural networks is basically the mimic is a simulation of the biological neural network which is there and the artificial neuron is called perceptron. So, in many book you can see it is call it is termed as perceptron. So, neuron or artificial neural.

(Refer Slide Time: 13:41)



Network is basically is the basic units which can solve many problems.

Now, let us see how we can mimic this biological neuron to our artificial neural neuron or it is called a perceptron. Now, here we can see that to this figure can be considering the two parts: in this first part we can see it is basically the figure of a biological neuron and the second part of this figure is basically.

So, the artificial neuron that is a perceptron now, here if we can see the input here in this artificial neural network $X_1, X_2, \ldots X_n$ are the input to the perceptron and all the input come to this part it is called the summation unit; it is basically same as the input from the different part it is coming like $X_1, X_2,$ $X_3, X_4,$ and coming to this part and this is the summation unit.

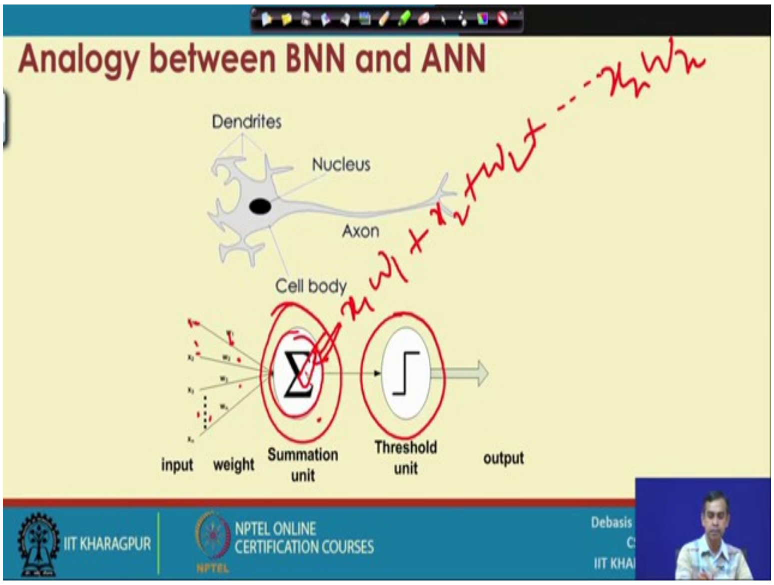
And another important thing that we can note it here is that. whenever the signal is coming here it, basically come with some weight W_1, W_2, W_3, W_4 it is like this. So, similarly it is here also the signal that is coming here with certain weights. Weights is basically indicates that how the signal is significant to this neuron? So, basically all signals those are coming they are called a weighted signal.

Now, when the weighted signal comes into this summation unit, basically all the signals and multiplied by their weights are summed up here and then total summation of this strength will be passed through this, this is just like axon this is just like a axon, and then come to this point and this point basically; now the signals which are summed up here comes to this point is basically same as the synapse or junction it basically connection to other neuron.

Now here the signals which are arrived here right will be check that; whether the signal strength is greater than the threshold value or not. If the signal strength is greater than the threshold value, that signal will pass to further, but if it is less than then it will not pass. So, so this way we can say this part is same as this part and this part is same as this part and this part is this one.

Now, so, this is the biological neuron and this is artificial neuron and we can see that how this biological neuron works it can be considered to work here and basically writing. So, far the program; that means, computation is concerned it has to computation. So, input is there and output is there as you know in every computation the input and output is there and this is a system which basically map given an input to a output and the mapping.

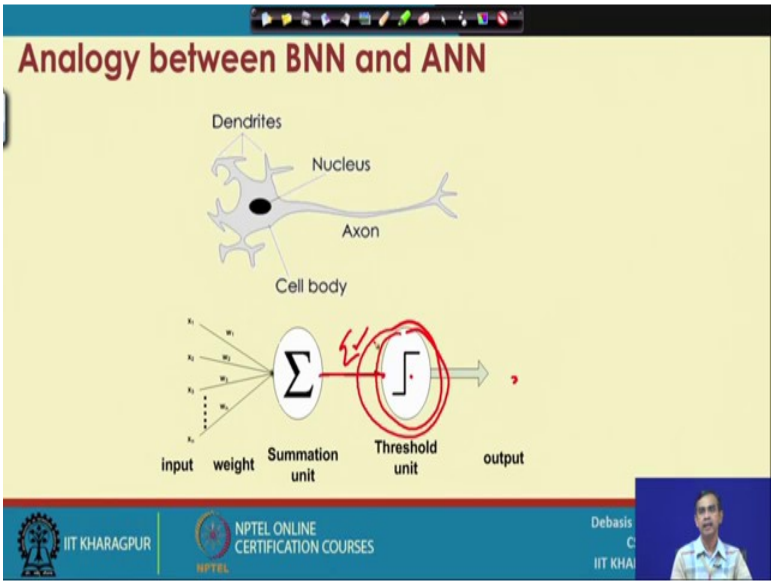
(Refer Slide Time: 16:55)



So, there are two mapping functions or simple functions are there, one function is basically; take all these inputs and their weights and the simple function that it will calculate is called the sum summation of products of all weights and their inputs; that means, X_1W_1, X_2W_2 and then sum of all these values. So, a simple program that can be written which take input X_1 and W_1 , X_2 and W_2 and produce

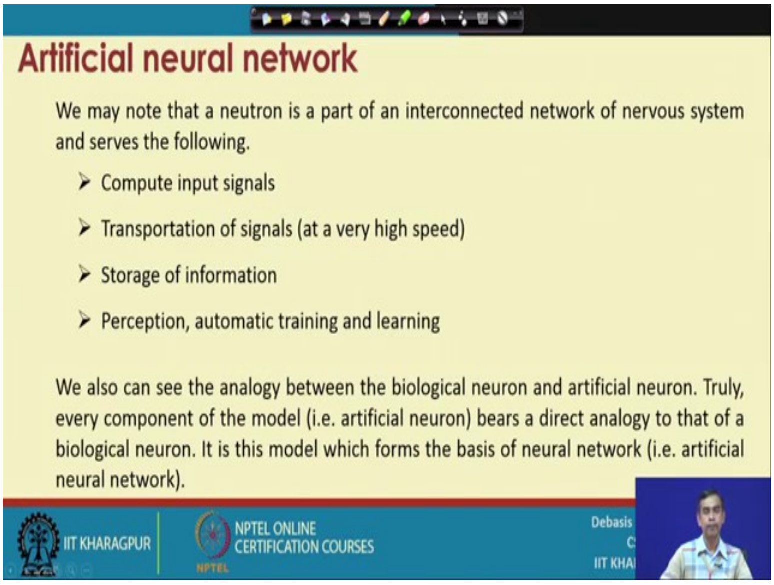
$X_1 W_1 + X_2 W_2 + \ldots X_n W_n$. So, this kind of so, this is basically computation that can take place here in this part a simple program with a simple loop can be right.

(Refer Slide Time: 17:40)



And then here one another program, we can think about whenever it receives this input; that means, these are sum of all the inputs it is there it will check with respect to some threshold value if the input this sum is greater than the threshold value then it will pass. So, it is basically what if then command is there a very simple code is there. So, what I can understand is that the way this biological neuron works we can write a simple program to mimic the working of the biological neuron by means of a perceptron.

(Refer Slide Time: 18:18)

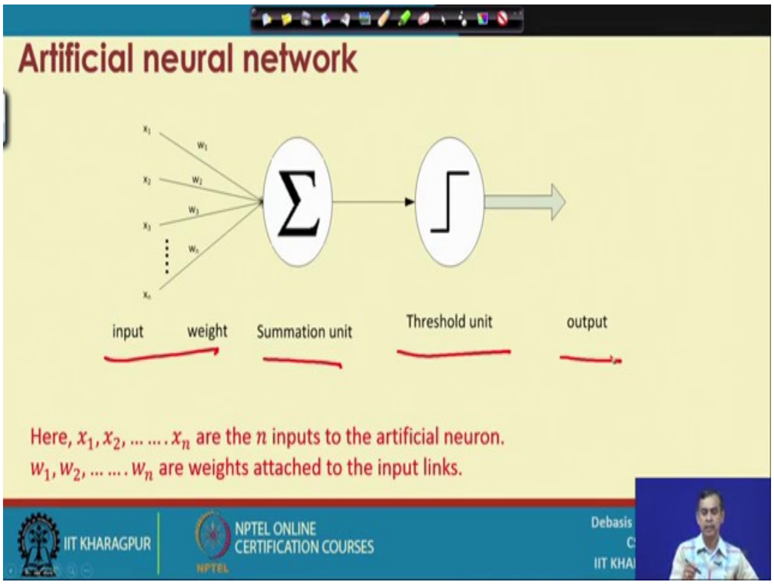


So, this is the idea the way the signal is work. Now few things are very much pertinent. So, far this perceptron and our biological neuron is concerned. So, as I told you a neuron is a basic unit and it works as an interconnected form. So, it is basically network.

So, that is why, It is got a network of neurons and this network of neurons computes the input signals, if you pass any signals as an input to this system. It will compute the signal and it can have the characteristic to transport the signals at a very very high speed and in addition to this, what is called the working of the signals few things are very important is that it can store information it can perceived and also it can learn automatically.

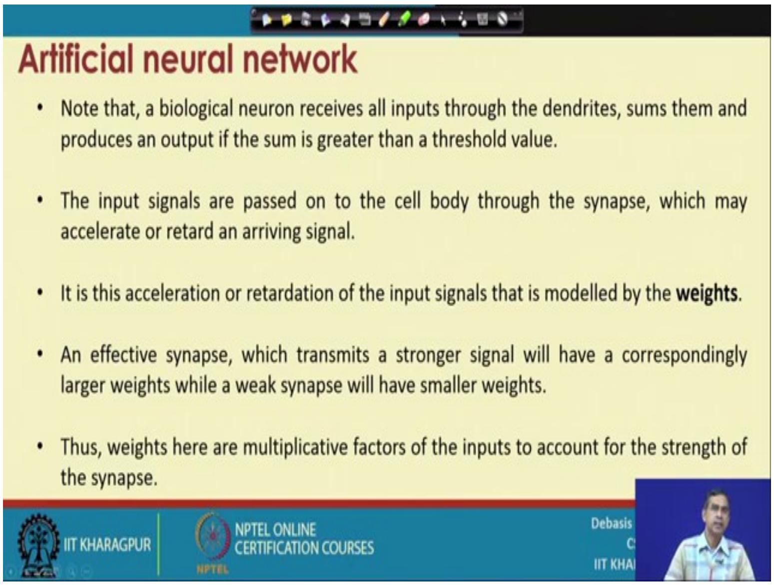
So, these are the concept that is there, and we will see how our artificial neural network the way the biological system works it also can be implemented and it basically give rise to the one important theory in the soft competing artificial neural network. So, this is the idea about.

(Refer Slide Time: 19:35)

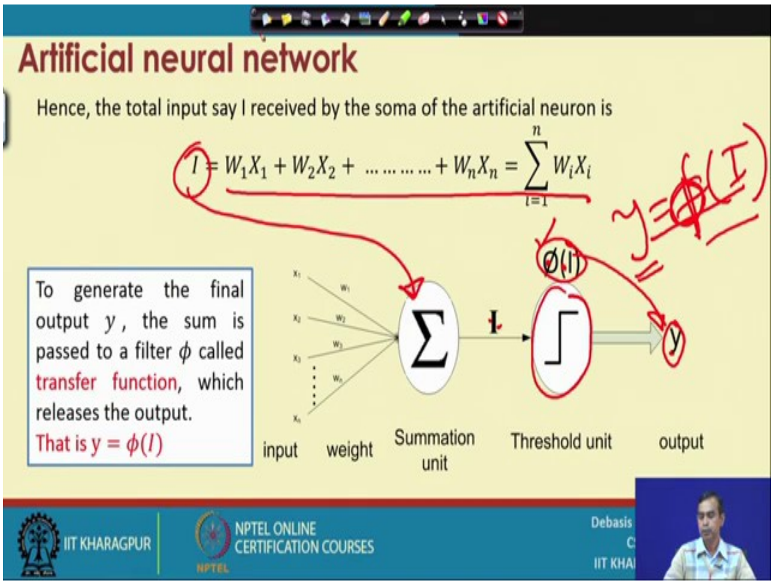


So, far the artificial neural network is concerned and as I told you that this work has certain computation per thing. So, input weight are the input and weight are the input and weight are the input to the things and this is one module or one function. Another function is output is there. So, so this way this neurons neuron system will work for us and now let us see. How this neural networks is basically solve many problems right there.

(Refer Slide Time: 19:59)



(Refer Slide Time: 20:07)

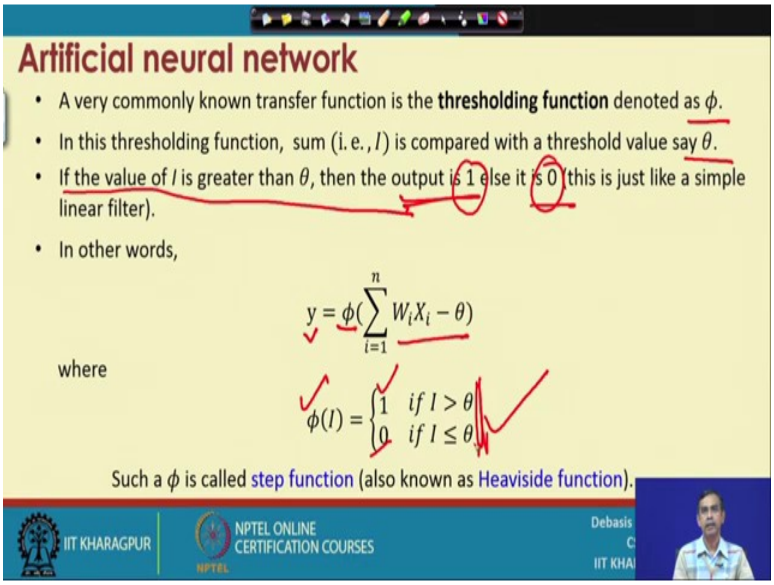


Now here exactly, again I just want to repeat the same thing, but in a different way. So, if this is the input like this is the input to the system, then reproduces the output by means of this program. So, I is that here this I passing there and here, basically function this function we called transferring function or transfer function this function is ϕ and for this $f(I)$ is the input and y is the output. So, this is the transfer function right. So, the this I . So, this function transfer function takes this I as an input and then produce the output.

Now, again so, this actually we can write y is a $f(I)$ or it is $\phi(I)$ like. As you know we have mentioned at the very beginning of this course any system has the antecedent and then consequence. So, it is antecedence and conjugate it basically maps input to output. So, mapping so, this way we can understand that this neuron or is a perceptron rather how map an input to an output.

Now, again in this processing one important thing that is there is called the transfer function. Now, we have to learn about the transfer functions and what is the meaning of this?

(Refer Slide Time: 21:34)

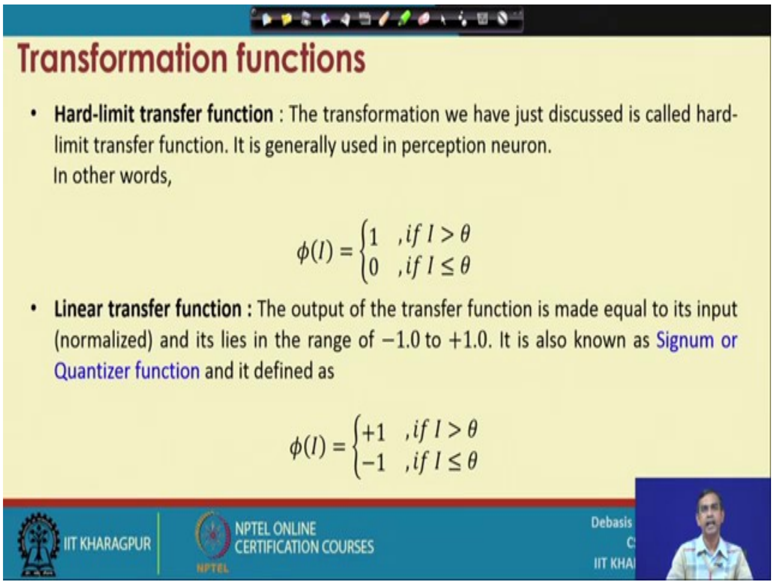


One here now there are in fact, many many transfer function known sometimes all these transfer function is also called thresholding function. We usually denote this transfer function as ϕ. Now all these transfer functions is basically compared the input I with respect to some threshold value. We denote this threshold value as theta.

Now the way this transfer function works is basically is a rule. That means, if the value of $I > \Theta$, then the output is 1, else the output is 0. Now, we will learn that the output of a neuron is either 1 or 0. It is not necessarily that always 1 or 0. Sometimes some other value also can be considered for, but for the sake of simplicity in calculation usually these two outputs are there. So, 1 and 0 so; that means, y has the value either 1 and 0. So, this ϕ returns either 1 and 0 and this is the rule that it follows. If I > Ɵ, then the function $\phi(I)$ returns 1; if $I \leq \Theta$, $\phi(I)$ is 0.

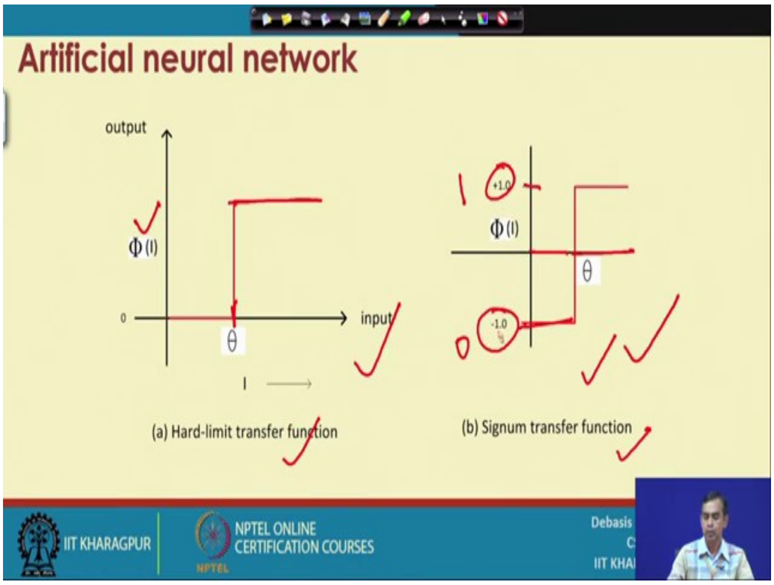
So, this is one transfer function that we have discussed and it follows the rule like this; and if a transfer function follows this kind of simple concept, then it is called a step function. Also, alternatively this function is called heavy side function. Now so, we have learned about the basic or simple transfer function that is there in the theory of.

(Refer Slide Time: 23:21)



Artificial neural network sometimes this the step function is also called hard limit transfer function. Other than this hard limit transfer function there is another function also known it is called the linear transfer function.

(Refer Slide Time: 23:36)

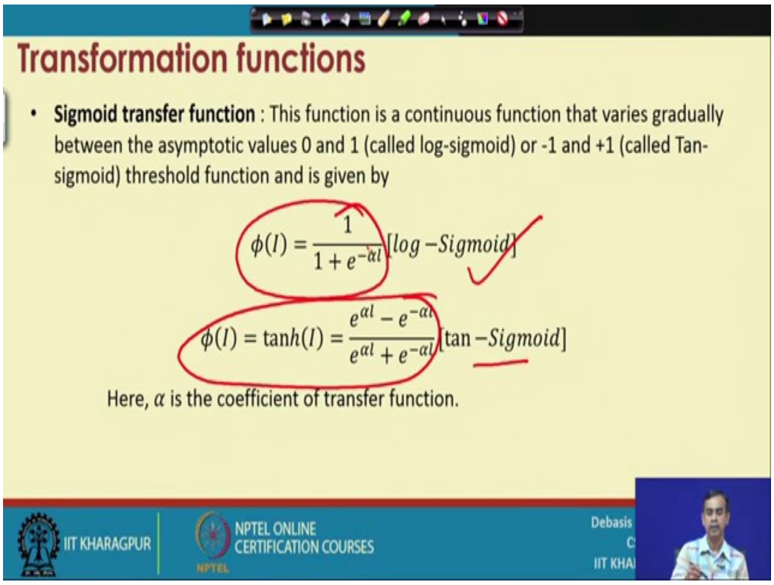


Now, here is the picture basically shows how the hard limit transfer function works and here is the Signum transfer function or linear transfer function. Now, in this case I can see that we see that, if the input is within this rang, then this function $\phi(I)$ written 0, and if the input is beyond this range then output that the function that returns is 1.

Now, this is the hard limit transfer function. On the other hand, Signum transfer function. So, it is basically if the input within these range it return -1 and beyond this range it will return 1. So, here in this case the output is -1 or +1. So, this is another one so, -1 also can be considered as 0, and this +1 also can be considered one if it is normalized to that one. So, anyway so, so Signum transfer function usually -1 and 1 hard limit transfer function, 1 and 0 although -1 to 0 two levels. So, two levels can be denoted by 0 and 1 also.

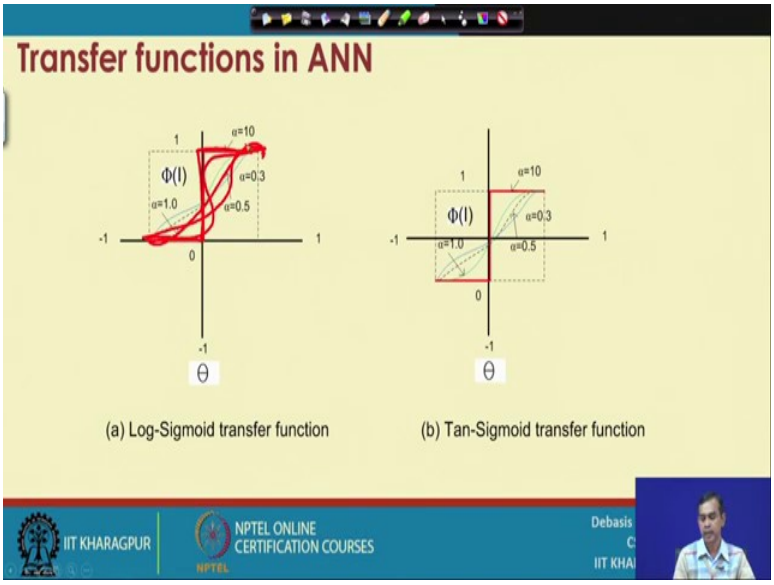
So, these are the two functions are there in addition to these two transfer function.

(Refer Slide Time: 24:53)



There are few more transfer functions are very important. These transfer functions are called Sigmoid transfer function. The sigmoid transfer function has two versions- one is called Log-Sigmoid function which basically takes this form and another is Tan-Sigmoid function which is basically take this form. Now, it apparently seems that these two transfer functions very difficult to compute, but there is a computation tricks by which all this calculation can be computed very efficiently, that we will discuss when we will consider the application of the neurons to solve problems anyway. So, we have learned few transfer functions which are very popular in the theory of neural network.

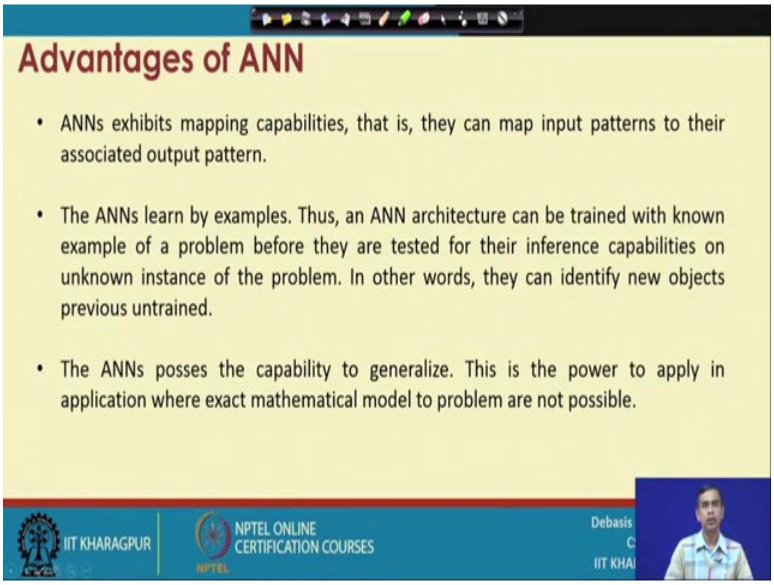
(Refer Slide Time: 25:39)



Now, after learning this transfer function. So, this is a graph actually. So, this graph basically shows how the transfer function that we have discussed just now. Log-Sigmoid and Tan-Sigmoid works is there. And here the different values alpha can be decided. If $\alpha = 0$, these basically same as the sigmoid function that we have discussed. If α value is 1.0 or 10 the sigmoid function will be like this. So, for the different value of this one the sigmoid function will take place like that.

Now the same thing is applicable to the Tan-Sigmoid transfer function, here the α , one important parameters right which basically decides, how the transfer functions will behave. Now so, these are the transfer functions.

(Refer Slide Time: 26:25)

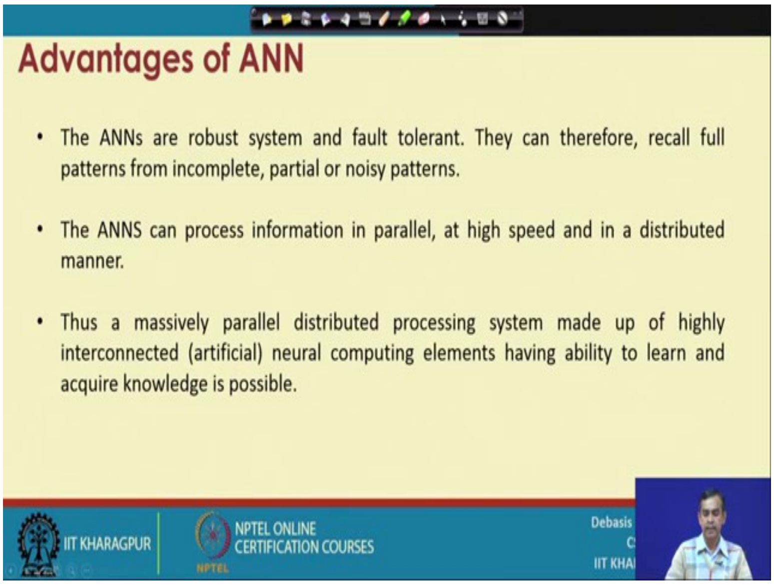


Now so, far the ANN is concerned why we should follow this ANN or the artificial neural network to solve our problem. This is because it has very nice mapping capabilities. That means, any input if it gives to you it can map to any output and that is with a very faster rate. So, that is why any input can be if it is pattern. Then it can read result the corresponding output patterns very effectively.

And another important thing is that; so, far this neural network is concerned, whatever the different parameters that we have mentioned the different parameters means the transfer function, the different parameter means alpha in the transfer function or the number of units or weights in the neuron all these are the parameters basically which characterized a behavior of a neuron.

Now, if we can decide the values of this neuron, then it is enough that the neuron can work for you. Now, again this values the all these weights, transfer function, the threshold values everything can be learned automatically if you trained the neuron. Now, we will discuss about how all these parameters can be learned automatically. Now, this is the one capability that the neurons are having. That means, automatically it can learn its value. And therefore, solve the problem. So, learning and everything will be discussed shortly, then we will be able to follow this concept.

(Refer Slide Time: 27:58)



So, this is our advantage and another advantage is a very much robust, fault tolerance. Therefore, it can recall full patterns for incomplete partial or noisy inputs. ANN can be used to process the information in parallel at a very high speed and in a distributed manner. This is, why these neural systems is effective for parallel distributed processing and we can solve any problems which cannot be solved using the single processing methods.

So, this is the advantage that the neural artificial neural network is having. Now so, we have learned about the idea about the basic units which is there in artificial neural network. And in the next lecture, we will learn about how this neuron can be trained to solve or learn itself for the different values in it.

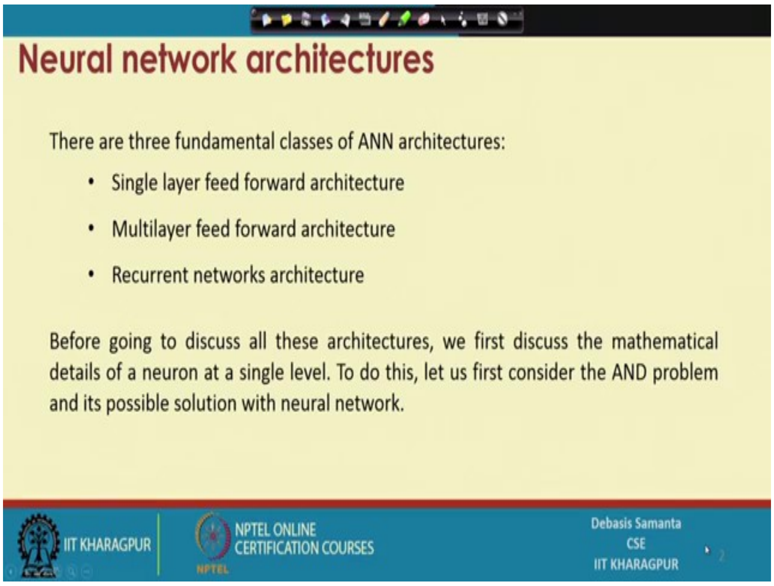
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 35
ANN Architectures

In Artificial Neural Network the basic unit is Neuron. And there are many neurons are interconnected to each other forming the network. That is why it is called the neural network. Now, today we will discuss in this lecture, what are the different architecture that can be used to build the neural network.

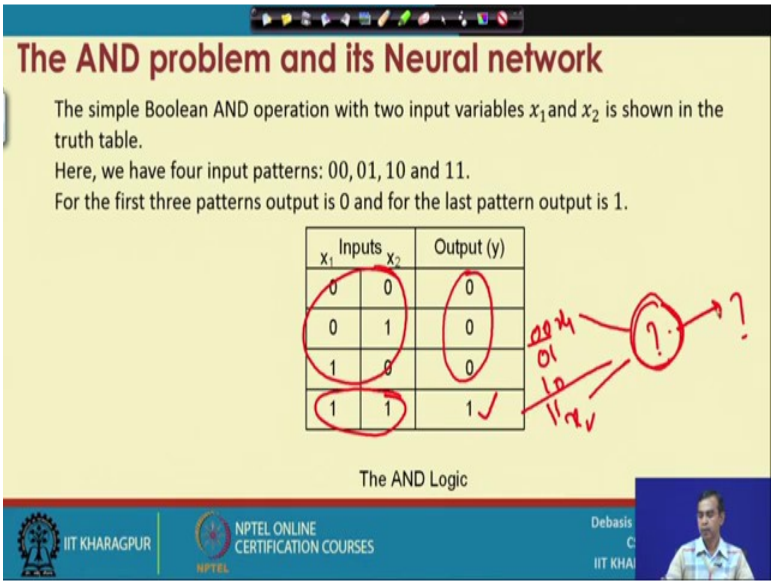
(Refer Slide Time: 00:34)



Now, all the architectures which are there in Artificial Neural Networks can be divided into broad 3 categories called the Single layer feed forward architecture, then the Multilayer feed forward architecture and finally, the Recurrent network architecture. Now, so, these are the 3 different architectures.

And then, before going to learn these architectures, we will just quickly go through the mathematical details of a neuron at a single level. So, to do these things, let us first consider a problem, we can term this is an AND problem and is the one problem is basically very much popular in Boolean logic and the AND problem like this.

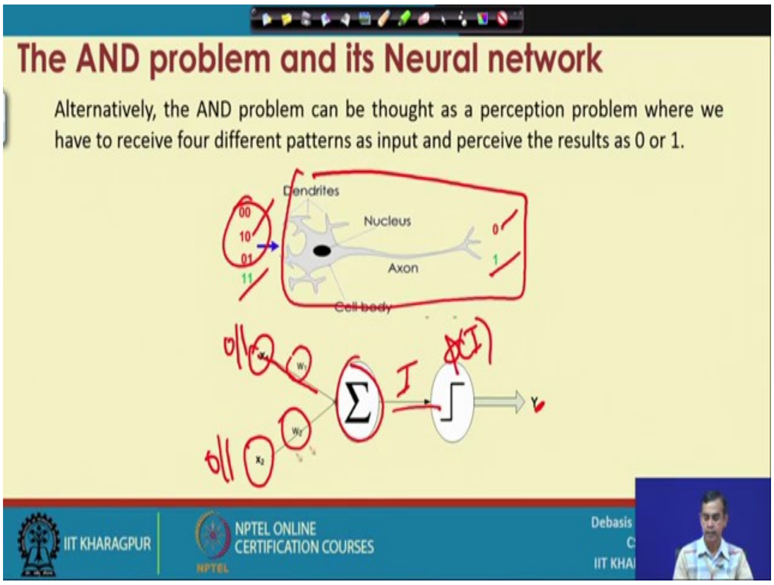
(Refer Slide Time: 01:25)



So, this is the truth table of an AND problem. Now, this AND problem can be considered as a Pattern Matching Problem or Pattern Recognition Problem. Now, let us see how it can be considered as a Pattern Recognition Problem. Now, we can consider if this is this is the neuron, that this neuron has the pattern consisting of two bits like say this one and this one so, x_1 and x_2. Now, these bits can be 0, 0 or 0, 1 or 1, 0 or 1, 1 ; that mean, x_1 can be 0 or 1. Similarly x_2 can be 0 or 1 and these are 2 bits are can be considered pattern. Now, what is the pattern recognition, that this you this system will do and it will give an output line.

So, pattern recognition problem like, so, if these are the patterns, if it is fed to this neuron, then it will give output 0. On the other hand, if this is the pattern, if it gives to this system, it will give that output 1. So, this way we can recognize the pattern whether it is 0, 0, 0, 1, 1, 0 or it is 1, 1. So, the that neural network can be right can correctly recognize this pattern either in the form of a 0 or 1. So, this is the one pattern recognition problem like and it is basically is nothing but a AND logic problem that is there in our Boolean algebra. Now, so, this AND logic and if this is a pattern, now let us see how the how the Neuron can be designed to solve this problem.

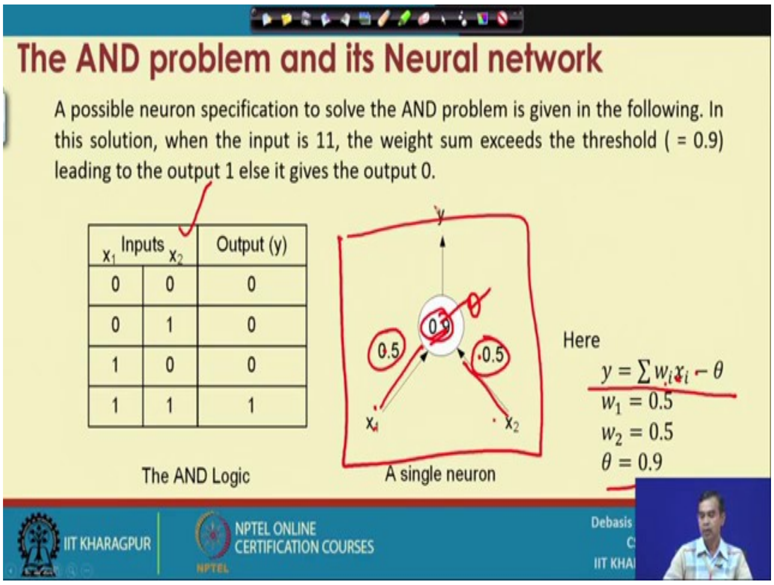
(Refer Slide Time: 03:10)



Now, so, here basically if we consider this is the biological neuron. So, these patterns whenever give to us like see if we see this pattern, we can say 0 and if we see this pattern, we can say 1.

So, is a Pattern Recognition. Now again, so far this mimic of this neuron is concerned; that means, in the perceptron, it has 2 input x_1 and x_2 and the summation unit and; obviously, w_1 and w_2 is there and it goes there and this is the transfer function ϕ, who is basically take the input I and then gives the output Y. Now, we will just see exactly what are the different weights values and then for this input? This means either 0 or 1 and here also 0 and 1. It can recognize the pattern. Now, so the idea about; that means, how the neural network is there to solve this problem, it is shown here.

(Refer Slide Time: 04:04)

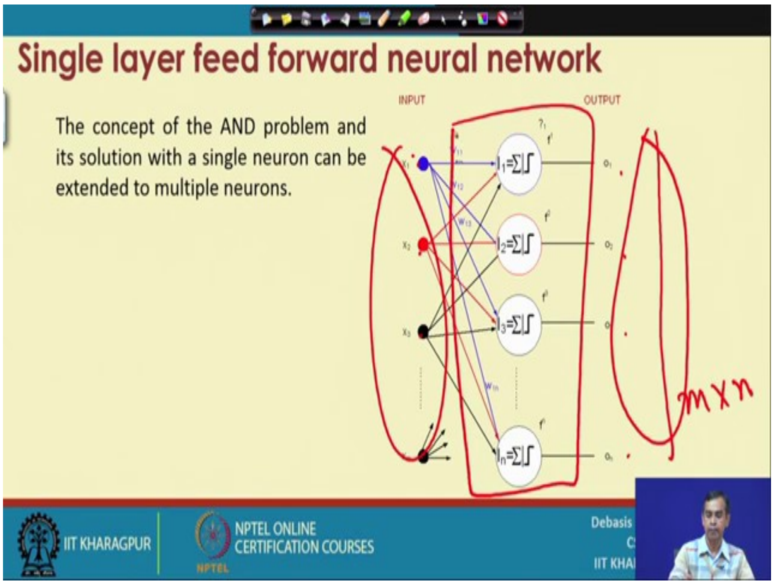


So, here we can see this is the patterns there needs to be recognized and this is our simple neuron or we can say this neural network consists of only 1 neuron and this is the only one neuron, it takes the x_1 is the input x_2 is the input and this 0.5 and 0.5 are the 2 weights in this case. And 0.9 is basically the Θ, the threshold values and then transfer function can be like this y, w_i, x y, it is visually $\sum (w_1 x_1 , w_2 x_2) - \theta$. θ is 0.9, then it will give the value either 1 or 0.

So, this is just like an idea about how this pattern recognition problem like this can be can be completed using a single neuron as it is shown here. Now, in the single neuron, so, few characteristics are important. These are the weights are there, we have to learn it. I gave you these weight values for an example, but you can ask that how these weights are calculated.

So, we will see how these weights can be calculated. Similarly, 0.91 threshold value, that we have discussed. Then, how this threshold value is known to us. So, we learn about how these threshold values is known to us. Now, this is the idea about how the neural it works for us.

(Refer Slide Time: 05:26)



Now, this is a simple one pattern matching problem; that means, 2 input AND problem that we have considered. The idea can be again extended, 3 input AND problem and so, on. Now, like this AND logic which is there in Boolean algebra, other logic like NAND, NOR all these things also can be implemented using this neural network. And you know, Boolean logic is basically, the basic things that is used there in VLSI circuit to develop our chips or computers processing unit computing processing unit.

Basically, it is the way it works, it is the same way it is also working here. That means, the way the size chips are basically designed. And basically same way our neural network can be designed to solve the problem. Now, so the last example that we have considered, it is on a very simple one AND problem, but if it is a complex problem; that means, if it consists of many inputs and then many outputs are to be considered, then that architecture will look like this. So, here we can see, these are many input x_1, x_2, x_3 ... x_n are to be fed into the computing system and it will produce O_1, O_2, O_3 and O_n, the n number of outputs are here.

So, it is basically m vs n_1 combination, that m input and n output is there. Now, so, this kind of input this kind of what is called a mapping from this input to this output can be managed by this kind of architecture. Now this is a one architecture if you see, here a number of neurons are stacked one after another. So, this is the one neuron, second neuron, third neuron and then nth neuron. So, if the number of output is n, then it basically requires n number of neurons, are to be stacked. Now, each neuron if we see. So, all the inputs that is are there is basically connected to all the neurons. So, x_1 is connected to this neuron, this neuron, this neuron, and this neuron.

The likewise, this neuron also connected to this neuron and this neuron. So, all inputs are connected to all neurons, which are there in this series. And another important thing that you can see also that, there are all the inputs are connected to this neuron by means of some weights are there. So, x_1 if this is the input to this one connected to this neuron, then it has weights $W_{11}, W_{12}, W_{13} \wedge W_{1n}$. Similarly, this is the input if it is there, then $W_{21}, W_{22}, W_{23} \wedge W_{2n}$. And for the n-th neuron, so, it is $W_{m1}, W_{m2}, W_{m3} \wedge W_{mn}$. So, this way all inputs are connected to all neurons and this is basically the network of connectivity that we can check.

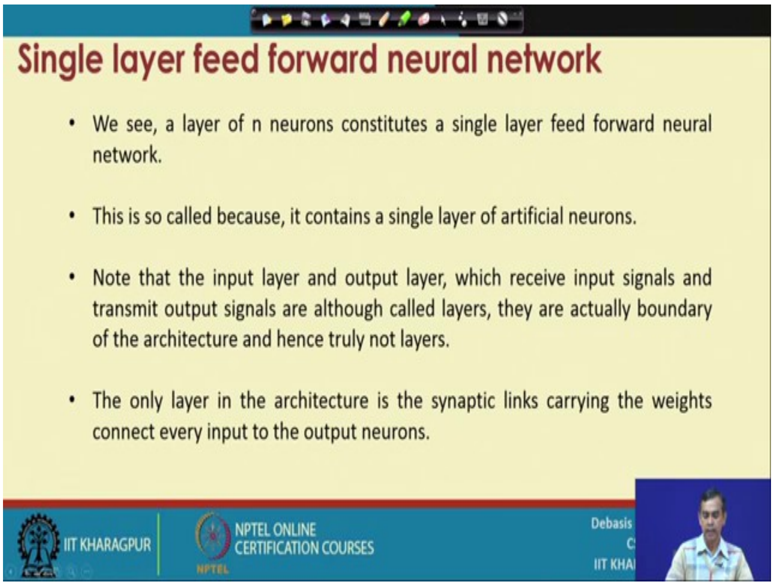
Now here, the input that is given to here will be feed forward to the output line. So, that is why it is called the feed forward neural network. And it is also called single layer; because if this is the one layer of neurons, so, that is why it is called the single layer feed forward neural network. And in this network, so, there are many weights. So, basically the weights that is there it is like this. So, all weights are there. So, they are maybe; so, weights is basically m cross n, a number of weights basically involved in this network and this is called single layer and then feed forward because of is the one layer. And then this input is connected to one layer only. And one important thing that I want to mention here again; so, that in each perceptron so; this is the perceptron 1 and this is a perceptron 2 and so on.

In each perceptron, there are what is called the thresholding function. So, it may be the different thresholding function or transfer function or may be all perceptron having the same transfer function. So, if it contains the different transfer function, then the learning that is required is very difficult. But if it contains only one transfer function, learning will be simple and straightforward. Further, in each transfer function there is a thresholding value. So, that threshold events also varies from one neuron or one perceptron to another perceptron. If we each perceptron contents the different thresholding value, then again also learning will be there and that learning will take much time.

So here, so further, Neural network is concerned and more precisely the single layer feed forward neural network is concerned, all the weights that is there to this layer are the parameters to be learned. And all the transfer functions that is there and with that thersholding values are to be learned. If we learn all and also number of what is called the perceptrons in the layer also one factors to be learned. So, if we learn for a given problem all these parameters, then we can say our neural network is trained perfectly. And once the neural network is trained then if we give any input to this it will produce the corresponding output. So, here only the matter of how a network can be trained or it can learn from the input to their output and then once it is build a network, we can use for solving our problem.

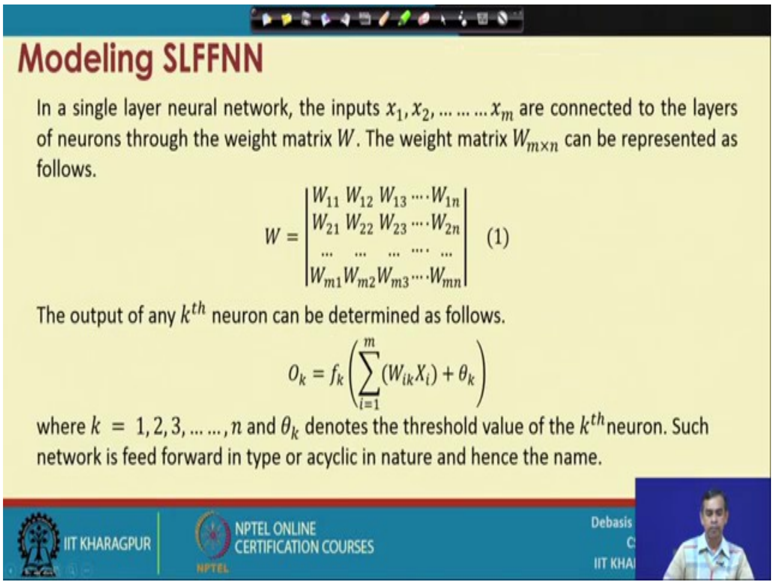
So, this is the concept that we follow in Artificial Neural Network. Now, we have started with first perceptron and we check that how AND problem can solve it. Perceptron is the very simple most on problem solving neural network. After that, just now we have learned about Single layer feed forward neural network.

(Refer Slide Time: 11:33)



Next, we learn about little bit different and complex whether neural network architecture, which is called the multiple Multilayer feed forward neural net.

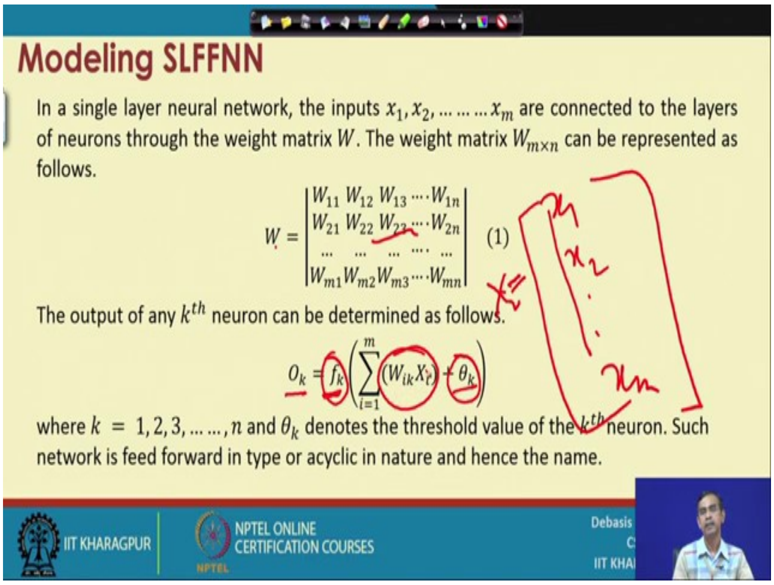
(Refer Slide Time: 11:37)



Now, again I just want to mention that I forgot to mention it there. So, in this particular neural network single layer feed forward network, the weight matrix W is there. It is basically the collection of all weights from any input to any neuron, any perceptron. So, these are the all weight values from the first input x_1 to all the neurons 1, 2, 3 to the n neurons. And see, this is the input from the nth weighted by these values are there.

Now, all these things is basically can be stored by means of a $m \times n$ matrix. So, it is called the weight matrix. Now, like this weight matrix, this is the transfer function that also needs to be learn and this is the threshold value that needs to be learn and this is for the k-th perceptron and it basically for the k-th perceptron these are the k-th transfer function with the threshold value.

(Refer Slide Time: 12:39)

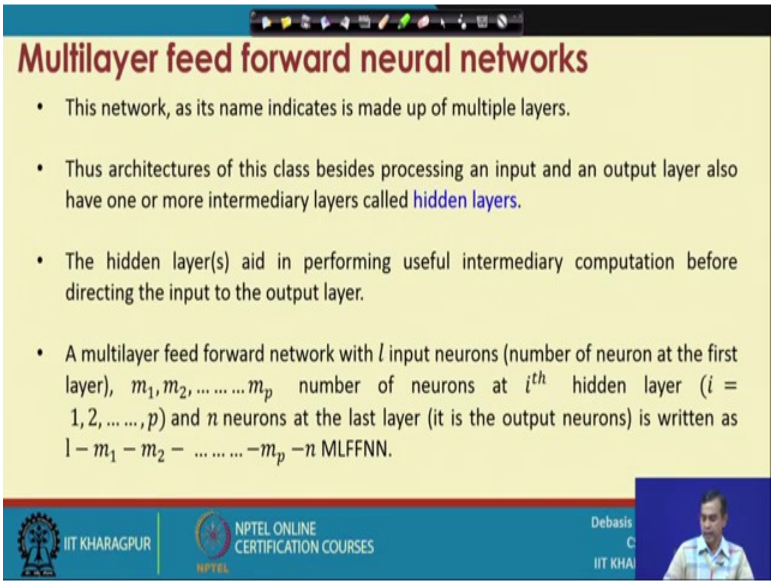


And given the input $x_1, x_2, x_3 \ldots x_n$ and that we can consider as a matrix X. So, any i-th input if it is given there. So, it is basically, $W.x$ is basically, the matrix product of the 2 weights and then this one, then it will gives the i-th what is called the summation unit to the i-th perceptron. So, this way it can solve and as you know the matrix operation is the one simple most and very fast operation.

So, computing in a neural network is very fast and then not a time timing issues are there ok. So, this is the idea about Single layer feed forward neural network. So, for modelling such a network is basically model this is the mathematical form or the mathematical model that can be considered or that is used to that is used to solve problem.

So, this is basically model of a Single layer feed forward neural network. And then, model can be expressed in a mathematically in terms of matrix and simple computation as it is here. So, so modelling is there and then. So, basically we have to build this model, means we have to find this is the W value and this is the value of this function. So, modelled means basically, we have to learn these W matrixes and f_k values, f_k functions for each neutron.

(Refer Slide Time: 14:09)



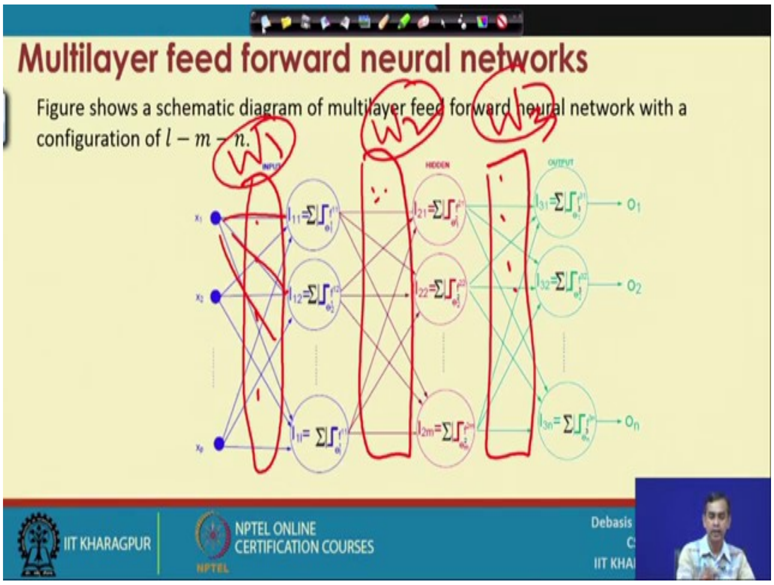
Now, so, this is the idea about Single layer feed forward neural network and we can extend the same concept to a little bit complex neural network. Now, in case of Single layer feed forward neural networks, only one layer of perceptrons are there.

On the other hand, in case of Multilayer feed forward neural network, instead of one there are many; out of which one layer which is connected to input data, it is called the input layer. And there is another layer which is connected to output is called the output layer. And in between input and output layer, there are some layers of perceptrons is called the hidden layers.

So, if l number of neurons in the input layer; that means, there are l number of inputs. If n numbers of outputs are there, then in the output layer, n number of perceptron should be there. And therefore, so, it is called the l and n combination. So, for the input and output is concerned and in between this input and output layer, they are may be m number of hidden layer say $m_1, m_2, m_3 \ldots . m_n$. So, these are the hidden layers. Then number of neurons in these hidden layers again can be divided by that number.

Say in m_1 layer m_1 number of neuron, in m_2 layer m_2 number of neuron and so on, so on. So, a typical look of such a multi-layer neural network can be shown here.

(Refer Slide Time: 15:34)



We say here in this figure, I show one simple layer, a simple Multi-layer feed forward neural network or 3 layers are there. So, this is the input layer, this is the output layer and this is the only one layer in the hidden layer. Instead of only one layer, are there may be like this many layers also can be considered. So, it will just increase the complexity of the network, that is all. Now, all the inputs are connected to this input layer and all the outputs are connected to be input layer. Here we can see, p number of inputs and n number of outputs and this network is called the Multilayer network because multiple layers of neurons are there.

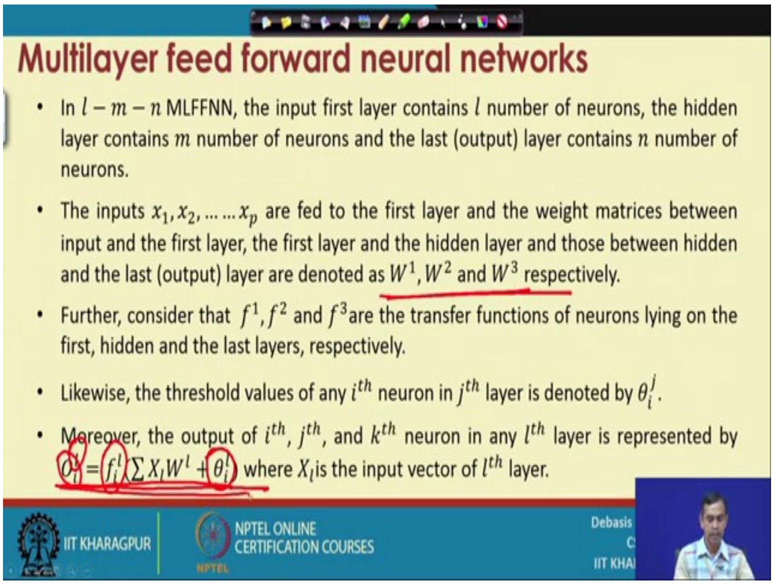
And it is also feed forward neural because input pass to this. It produced the output; whatever the output produced by a particular perceptron, gives the input to all other perceptrons into the next layer. Then this output also takes this input from the different perception in the previous layers and produced the output gives to the this one. So, it is the same thing. So, it is basically in each layer, we can say it is a single layer line. So, there is a what is called the stack of a number of single layers.

All are highly connected to each other and then it forms the Multilayer feed forward neural networks. Like the modelling of Single layer feed forward networks, the Multilayer feed forward networks also can be modelled in same way; but here, more what weight matrix, more transfer functions and the different thresholding values are there.

For example, so, there will be one weight matrix is required to define all the weights here, another weight matrix is required to denote here, another weight matrix is required to denote here; because all the inputs are passed through the different perceptron by via the different weighting values there. So, here you can say the weight matrix w_1, here w_2 and w_3. So, in order to model this one, we have to know what is the weight matrix, what is the weight matrix at the different level. So, these are the modelling issues and then for each perceptron in each layer, the transfer function. So, it is $f_{11}, f_{12}, f_{13} \cdots f_{1p}$ like this one. Similarly, here also this kind of transfer functions are there.

And in each transfer function, thresholding values also to be considered. So, in order to model such a neural network, it basically modelling these are the weights and the difference thresholding transfer function and then threshold values in each perceptron. Now, so all these things can be done again in a simple, in a compact and mathematical form by means of different weight matrix and then different what is called the functions that it can conserve. For example, this is the weight function that can be generalized to represent each perceptron in each layer.

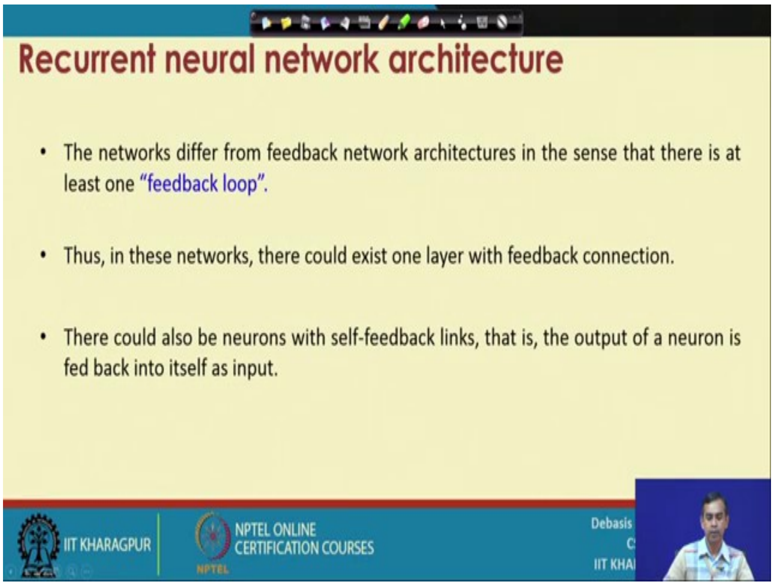
(Refer Slide Time: 18:17)



So, it is basically the output of the i-th perceptron l-th layer and it is defined by the f^i function, the transfer function and this is the threshold function. So, this is from this is the unique values unique functions in each perceptron that is there in addition to these are the different weights that needs to be considered there.

So, modelling just like a Single layer feed forward network, if it was modelled using only one weighting matrix, it will be modelled using 3 different weighting matrix. If it is modelled by a series of transfer function and threshold value, but it will be a large correction of transfer functions and thresholding values. So, modelling only will be little bit complex than the single layer feed forward in case of Multilayer feed forward neural network.

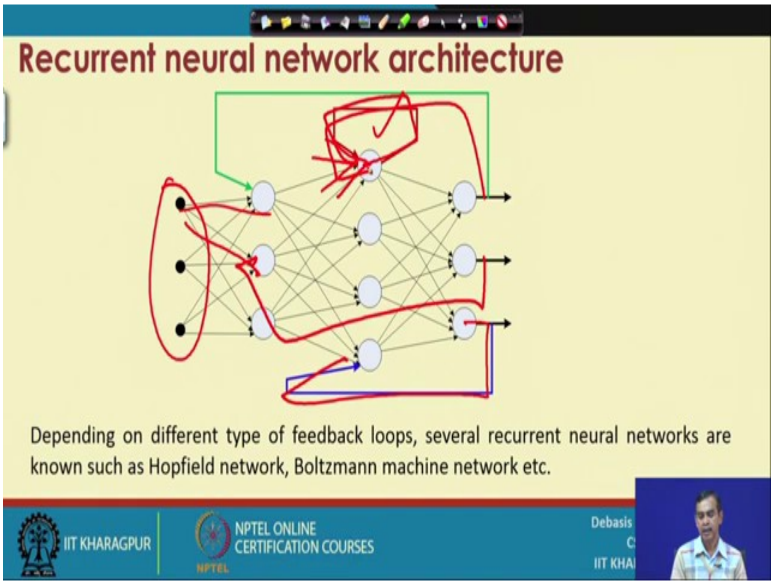
(Refer Slide Time: 19:23)



So, this is the idea about Multilayer feed forward neural network. Next type of neural network that it is called Recurrent neural network architecture. Now, the difference of this network architecture compared to the previous two architectures is that the feedback will be there.

That means, there will be a loop. So, it is called the feedback loop. So, there may be exist one feedback, a feedback loop from the next layer to the previous layer. So, if feedback is there, then it is called the Recurrent neural network. And now, so, let us see how the such a network looks like.

(Refer Slide Time: 20:05)

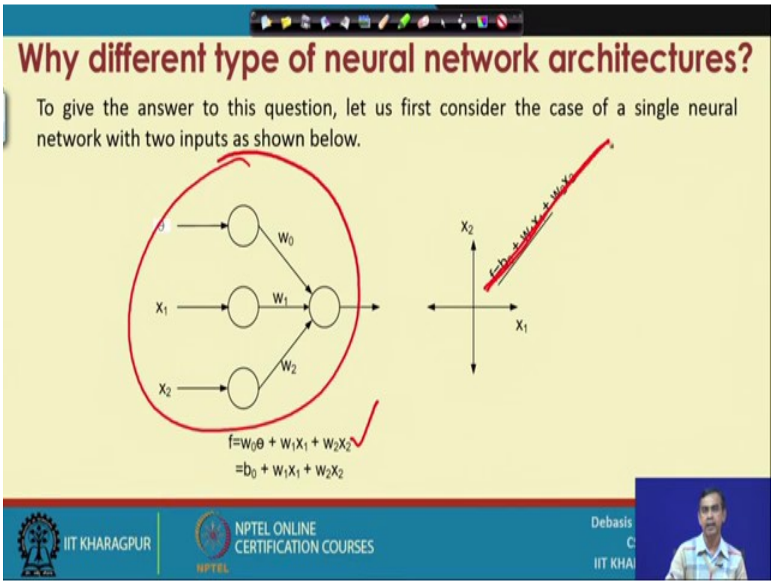


Here is the one picture here. So, here basically if we see, the output of this is feedback to this one. The output can be feedback to this one also. And here also output to the same neural this is called a self-loop and this is the previous loop that; that means, it is more complex because in addition to the conventional input there which will be connected plus the output from any near to the previous perceptron is there.

So, number of what is called the output to each will number of inputs to each perceptron will be enormously high. And it does leads to a very complex network architecture called the Recurrent neural architecture. So, if there is a self-feedback or recurrent things then they are call the Hopfield neural network, Boltzmann machine network like this on. So, different networks are there. Whether there will be a self-loop or not, the loop from the next layer to just previous layer not or the loop from one perceptron in a layer to any perceptron in any other layer or not.

So, this way the different architectures can be thought and then it is there. Now again, so, for the modelling is concerned the same concept, it can be applied here and then same modelling will be there. Only the thing is that, all the matrix that is there, they will have very larger size compared to the simple feed forward neural network.

(Refer Slide Time: 21:39)



Now, so, this is the Recurrent neural network architecture and there is the; obviously, the question is that, which network architecture is suitable to which application? Now, we quickly gone through this concept there, in which case the Single layer network is required in which case the Multilayer network is required or in which case the Recurrent neural network is there. Now here, let us consider this figure. In this figure we can see so, this is the input layer and this is the one processing on output layer. So, neuron is there. So, it is just like a Multilayer feed forward network sort of thing or we can say this input is directly come to here this also we can consider then this layer is not there.

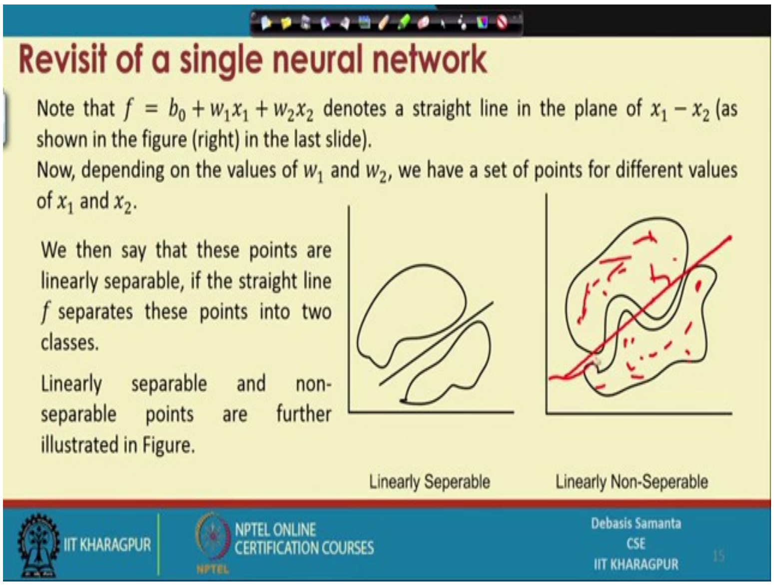
So, it is basically a Single layer feed forward neural networks are there. We will consider 3 weights; w_0, w_1, w_2 and this Θ is called the bias input sometimes use there. Mainly there are 2 input; $x_1 \wedge x_2$ and this can be x_0 also can be written like there. Anyway, so, if this is the neural network, then we can say the transfer function will be look like this. So, basically the transfer function basically summation of the I input into their weights plus this is the threshold value.

So, this is basically nothing but the threshold function it is there. Now, so, this basically f will return depending on the different values of Θ and weights it will return one output. Now, this basically if we see, this is this is an expression of a straight line in a two dimensional phase $x_1 \wedge x_2$, if we consider the two dimensions there.

So, it is a two dimensional data space, $x_1 \wedge x_2$ and for any input values having $x_1 \wedge x_2$, it basically decides either this input is in these sides or in these sides. So, this basically way. It basically classifies the data it belongs to these sides or not these sides.

So, it is also a classification or is a we can say prediction concept like. And this kind of prediction concept we can see how it can be solved using this Single layer feed forward neural network. And in this case, if this kind of expression is possible, then this is basically straight line. So, here the predictor or the classifier looks like a linear and straight line is there. Now, so, this is a linear classification or linear prediction is there.

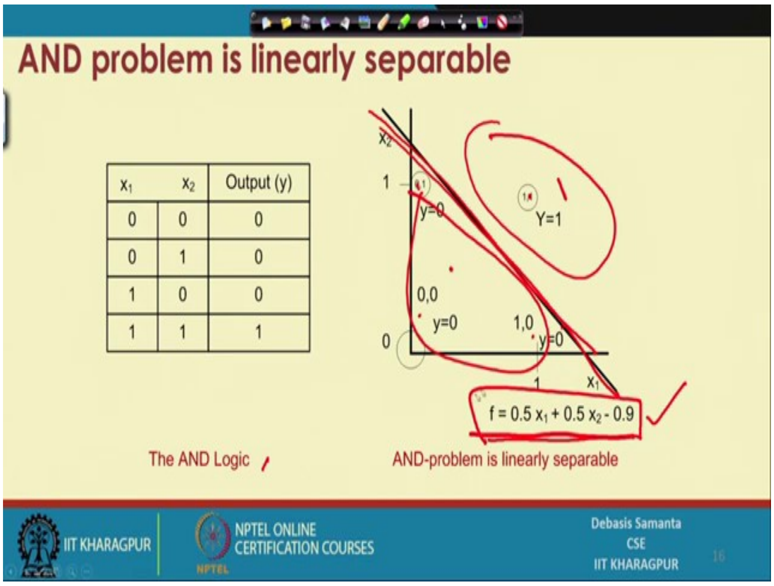
(Refer Slide Time: 24:09)



Now, if these kind of predictions are there, then we can implement these kind of things using a Single layer forward neural network. But there are some classifications are there, which where the data cannot be linearly separable. Now, exactly I can discuss about what is the linearly separable data is there? Now, so, suppose these are the data and these are the another type of data, so, 2 patterns; one pattern is this one and another pattern is this one. Then we can think about a linear line to separate all the patterns into 2 parts. So, this is the one pattern type, another pattern type. So then, we can say that these data are linearly separable, but like here if this is the patterns like this and these are 2 different patterns, then no linear line can be thought of.

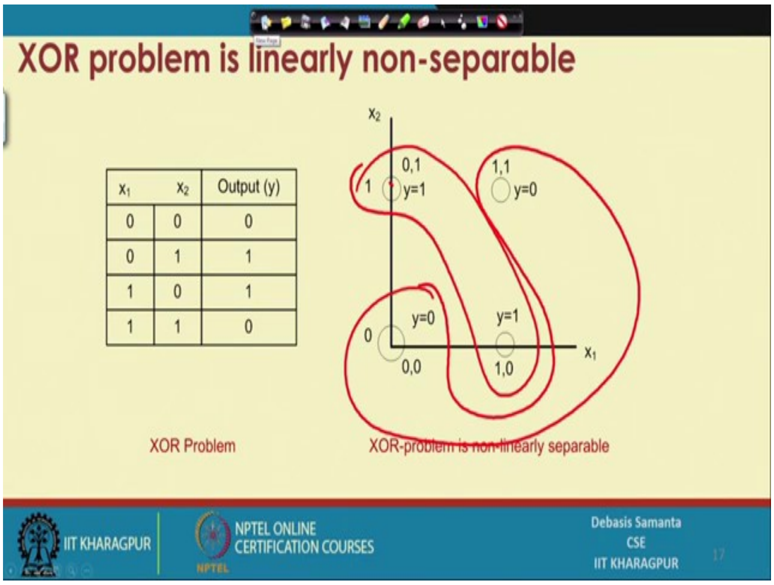
So, it is basically this is a very difficult to pattern by a linear line. So then, we can say that these are the data are linearly not separable. So, there are 2 types of data. So, for the prediction or pattern recognition is concerned data is very linear; that means, it can be separate linearly and something is cannot be separate non-linearly. So, if the data is linearly separable, then we can use simple network the Single layer feed forward network. But if the data are not linearly recognizable or separable, then we should consider the network which is other than Single layer feed forward network; that means, Multilayer or Recurrent near like this one. Now, so, there that so, these are the basically rational or genesis that in which situation, we should follow which kind of neural network architecture.

(Refer Slide Time: 25:42)



Now, here is an example that, I want to give it. If the AND logic that is the case, we have discussed about. So, this is the AND logic has that pattern. These are the different input and this is the another output. So, all the input can be patterned as a 0 and all the input in this size can be pattern at the 1. So, 2 patterns and there is a straight line like this one, which is the equation of this form that can be used to separate this one. So, this is the neural network and implementation of this neural network by means of these kind of lines and then these data are linearly separable data and then a Single layer feed forward network can be used to train to develop this model.

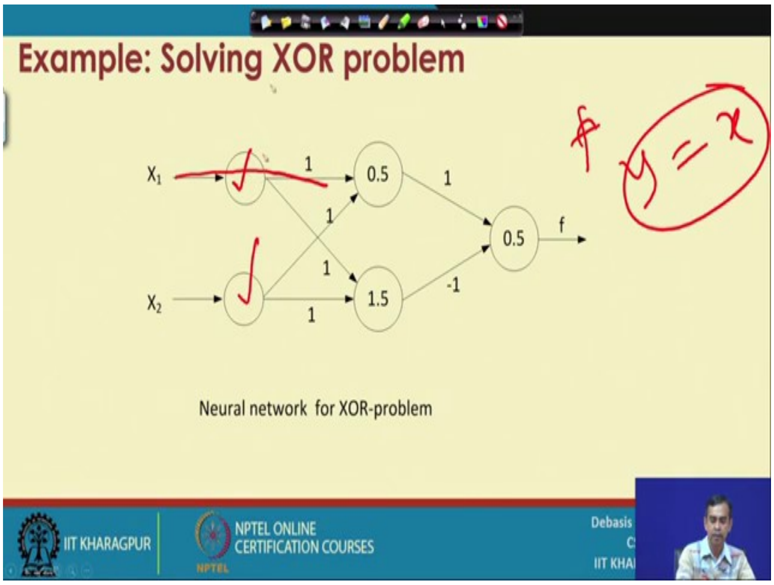
(Refer Slide Time: 26:25)



On the other hand, let us consider another problem. It is called the XOR problem. So, XOR problem has the pattern like this wherever the output will be like this. Now if it is like this, so, different patterns it is there. So, here basically, so, these are the one pattern and these are the another pattern. So, these are 2 patterns. Now, we can see that if this is the 2 patterns given to us, the data cannot be linearly separable, that data cannot be linearly separable means, we need something else then the single layer feeder. So, in this case, we can follow the Multilayer feed forward neural network. And let us see how the Multi-layer feed forward networks can be planned to design this kind of architecture. So, their problem is like this.

So, it is linearly non-separable.

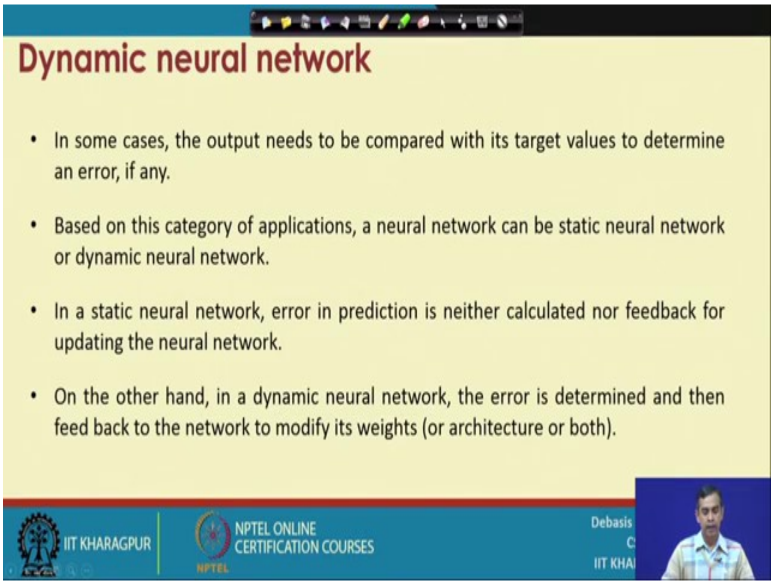
(Refer Slide Time: 27:12)



And here is architecture of the XOR problem. So, this is a one layer, the hidden layer and the output layer; input layer, hidden layer and output layer. So, it is a 3 layers. In this 3 layer 2 inputs are there and 2 perceptrons and in case of hidden layer, again 2 perceptrons, in case of output layer, 1 perceptron. Now, here the different weights, here the different weights. Here, we can say the weights are this one and here the different weights we can consider. So, there are 3 weight matrix are there and here the 0 or 1 threshold value you can consider. The simple linear transfer function it has directly pass it; that means, y = x.

This kind of transfer function it is here. So, we have written blank. It is this is y. Now here, the different transfer function is to be followed. We will come to this and then the threshold value that can be considered in each unit, it is shown here in the hidden layers and also through some value that can be considered here, it can be shown here. So, this basically the architectures which basically takes any input pattern and then produce output occurring the XOR logic and this kind of things is possible only. So, XOR logic cannot be designed using Single layer feed forward network. In fact, no one can design it. So, it can be designed only using this kind of simple Multilayer feed forward neural network.

(Refer Slide Time: 28:32)

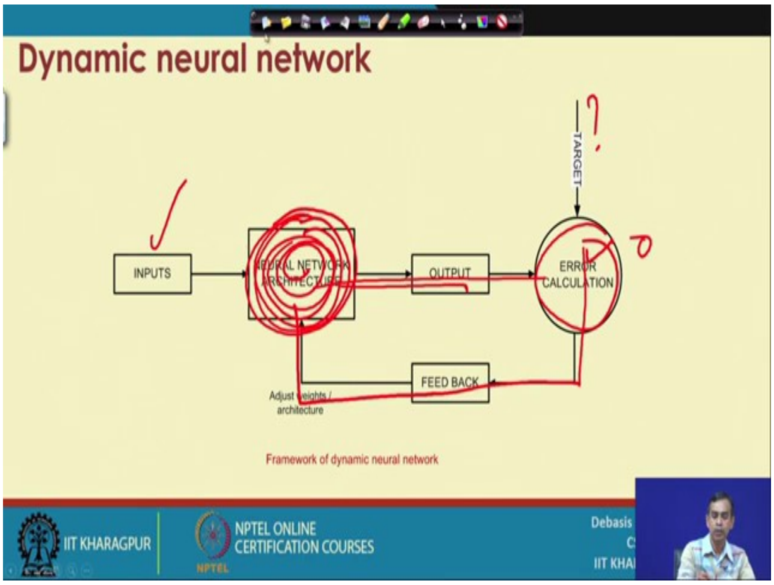


Now, so, these are the simple network that we have considered. Other than this network the Single layer, Multilayer, there is another one network also known in the theory of Artificial Neural Network is called the Dynamic neural network. Now, Dynamic neural network come into the way that if at any instant, if we decide that this is the output and then we can calculate its error; that means, output should be x and it is coming at x' then the error is $(x' - x)$.

Now, if we it this error to configure our neural network automatically, then it is called a Dynamic neural network. Now, so, there is a Static neural network versus Dynamic neural network. In case of Static neural network, no error computation takes place. On the other hand, in case of Dynamic neural network, error needs to be calculated at every instance and based on the error. That we have obtained the network parameter can be adjusted, then it leads to another neuron into or it is called the Dynamic neural network.

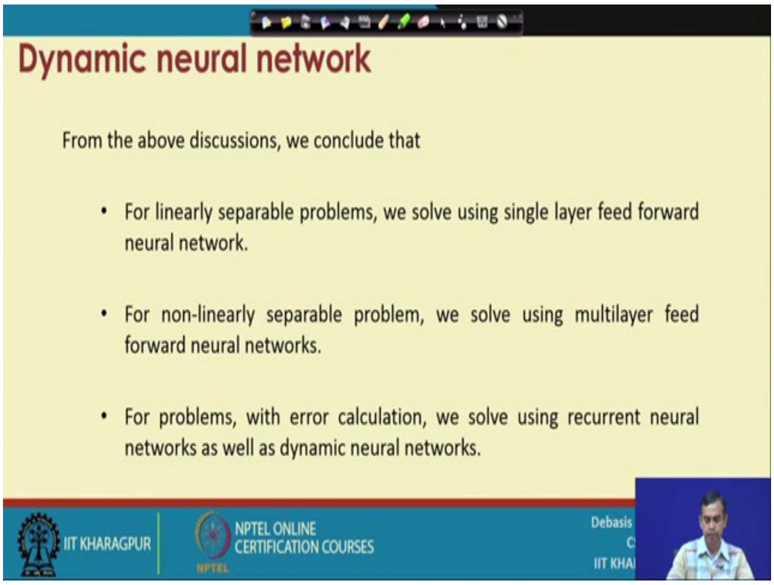
So, you can understand that it is not a; obviously, the feedback will be there. It is just Recurring neural network type, but error needs to be considered as the feedback in each neuron. So, it is the concept and it is; obviously, too much complex. So, for the network architecture is concerned and so, further computation is concerned because in every instances, we have to calculate the error and if error can be propagated back to the previous layer neurons and then that neurons then adjust it weights or parameters values, so, that the error can be minimized.

(Refer Slide Time: 29:57)



So, this is the pictorial description, if the input is given to these and this is the Dynamic neural network architecture. Output will give there and there is an error calculation unit which basically recalculate the error and give a feedback to this one. And using this feedback, this neural network will automatically update or dynamically update so that, the error can be minimized or the 0 error or the target output can be precisely obtained. So, basically this an automatic adjusting this neural architecture will take place and then neural net will give the best output and in that case it is very fault tolerant or robust system can be developed. So, this is the concept of Dynamic neural network.

(Refer Slide Time: 30:55)



Now, so, as a summary you can see that, for linearly separable problems we can designed the Neural Network Architecture in the form of a Single layer feed forward neural network.

On the other hand, for non-linearly separable problems, we can use either Multilayer feed forward neural network or the higher configuration likes in a Recurrent neural network or Dynamic neural network. And Dynamic neural network in particular can be used for error calculation, we solve and then Recurring neural network and the Dynamic neural network, we can use for the if we want to take into account the errors in the computation. So, these are the different architecture that we have learnt and next our idea next our task is basically how to model the different network architecture.

So, modelling the different networks architecture means, how to learn the different parameters with which a neural network can be composed of. So, that will be discussed in the next lectures.

Thank you.

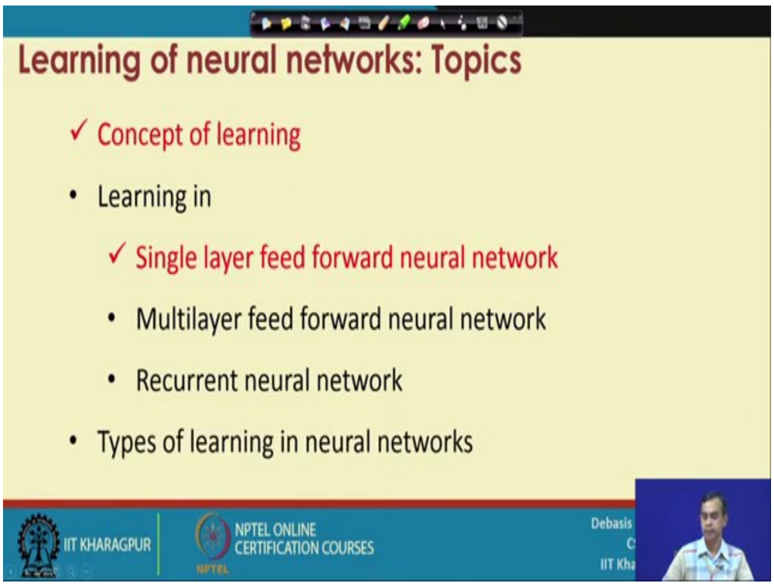
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture - 36
Training ANNs

We are discussing about solving problem in Soft Computing using artificial neural network. In the last two lectures, we have learned about how the basic unit which is there in our human body. So far the nervous system is concerned, that is the neuron can be mimicked to a perceptron. And then we have discussed about how such a perceptron can be modeled to solve problem. We have also discussed the different architectures that can be used to solve different problems or varieties, varieties in case of varieties of complexities and the different architecture we have discussed. And so for the neural network is concerned, the architecture is basically the first thing that we have to decide to solve our own problem.

Now, in order to discuss this architecture, we have decided that how an architect how an architecture can be modeled. Now modeling an architecture is basically by means of weight matrices and then the transfer functions and a threshold value in it is perceptron. In the next, in this lectures we will learn about the idea about how the learning can be accomplished and then what are the different learning techniques are there.

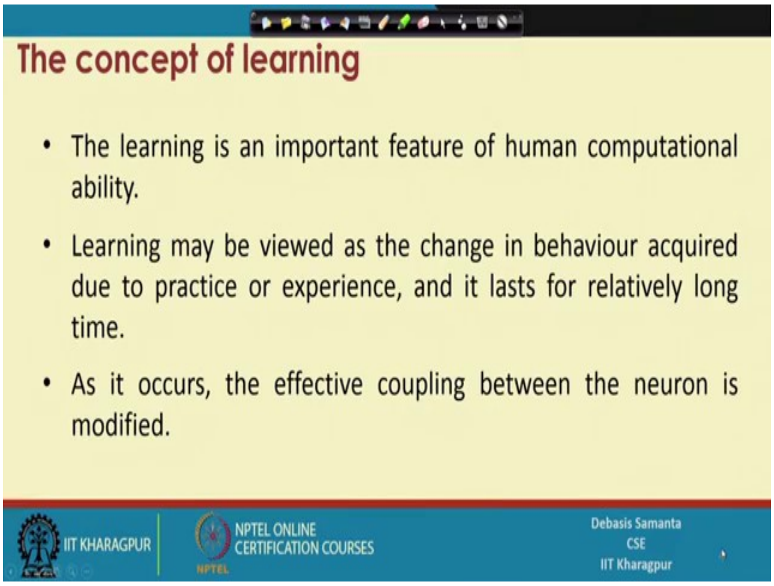
(Refer Slide Time: 01:32)



So, basically first we will discuss about the concept of learning and then as the learning as the architecture is different than definitely, the learning concept also will be different. So, in this lecture, we will discuss about after concept of learning how to learn a single layer feed forward neural network and then multilayer and then other neural network learning will be discussed in the next class, next lectures.

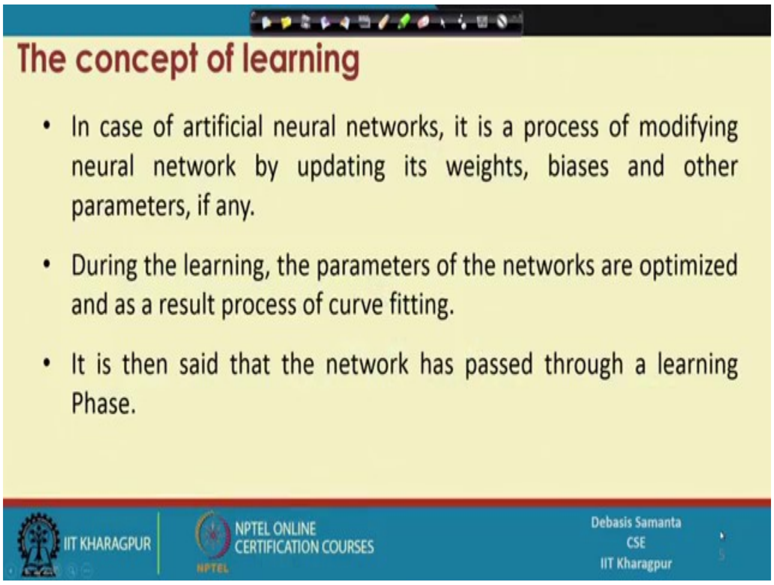
Now, so first let us understand about what is the concept of learning. Now concept of learning here in our every daily life, whenever we are learning every day in fact, whenever we see something, from there we can learn many things. So, basically we see, we here, we can sense and therefore, we can learn. So, the learning it is inherent in our body, in other human or in any other living things; even trees, the animals everybody is in fact learning from the environment, from the atmosphere where they belong.

(Refer Slide Time: 02:40)



So, it is also the same thing is applicable in case of our neural network. The learning is also there and the learning means basically in case of our neural network. Learning means how the different values of each perceptron can be learned. So, this learning is basically the idea it is there.

(Refer Slide Time: 02:58)



And in case of artificial neural network, it is basically the idea is that it is a process of modifying a neural network by means of the different weights, the threshold values and other parameters namely the number of what is called the perceptrons, number of layers and then the threshold values and may, so many things are there.

So, basically when we can train one network artificial neural network it is also called when artificial networks learn. This basically they try to find optimum values of all these, I mean how we can find an optimum value in terms of number of perceptrons to be included in the network, how the network connections can be optimized, how the different threshold values can be decided and all these things are there. So, if the network can learn it by means of some methodology then we can say that the training of the network is over. So, the network can learn itself from the training data.

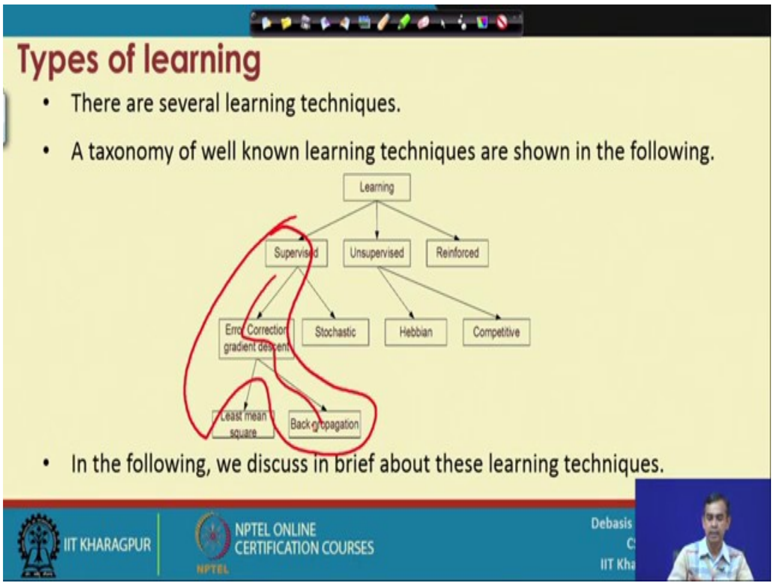
Now, here the idea about that how network can learn it in order to learn basically how we the human people can learn it. So, for human learning, we need a large number of inputs or in other word, we can say a teacher is there; teacher can say many thing and this is the input. So, if the input is known to us then we can learn from the input. So, here also in the neural network, the input is needs to be fed to the network and then from the input, neural network automatically learns it.

So, this input is called in neural network learning is called the training data. So, once the training data is known then we will be able to learn the network effectively. As an example, I can tell this say suppose we have to recognize handwritten character. Now, so far the handwritten character and we want to use this neural network to solve this problem.

So, what is the training data in this case. I can consider 500 subjects. So, that they can write the different characters in the English alphabet, suppose a the different people allowed to write the A. So, 500 people in this case. So, 500 A’s is are there. Now this 500 A's can be the training data. So, if we give this training data to this neural net, then it will learn automatically to recognize the different the characters. So, for example, A character first, then B character, then C characters and so on, so on. So, batch wise the different characters or in a single batch, all the characters with different singles can be given and then the neural network can recognize this pattern as either it is A, B, C or this one.

So, this is the idea it is there. So, training data maters. So, training data from 500 or training data from 50 people or 50 samples that can be fed to the neural network and if we give it to the neural network, it automatically decides it is how many perceptron is required to solve this problem and how many weights and weight matrices are there, how many layers are there and what are the transfer function that can be considered and what are the threshold values that can be considered. So, this is the concept of learning there in the neural network.

(Refer Slide Time: 06:25)

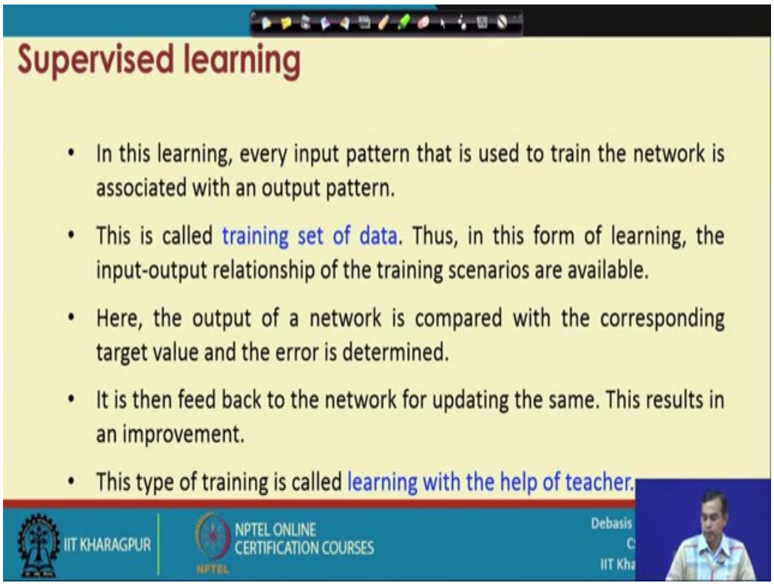


Now, so far the concept of learning is concerned, the theory of learning is very exhaustive there. In fact, there are many learning principles, many techniques are there. Now in this slide, we see some learning techniques. All the learning techniques can be classified into broad 3 categories called the Supervised learning, Unsupervised learning or Reinforced learning.

Now, supervised learning can be again decided as a stochastic learning or error gradient based learning, error correction gradient descent learning method. And then error correction gradient descent method again either least mean square, least mean square method or back propagation method. On the other hand, Supervised, Unsupervised learning; there are 2 types, one is called the Hebbian learning technique, another is Competitive learning technique.

Let us quickly have some brief idea about all these different learning techniques and then as it is the constant, the timing constant is there will not be able to cover all the learning techniques. But, only few learning techniques will be covered. More precisely, we will discuss about Supervised learning and then the Back propagation learning this one. So, the we will limit our discussion to these kind of things are there. All other techniques, we should have a little bit brief idea about the overview, not the details concept of the learning it is there.

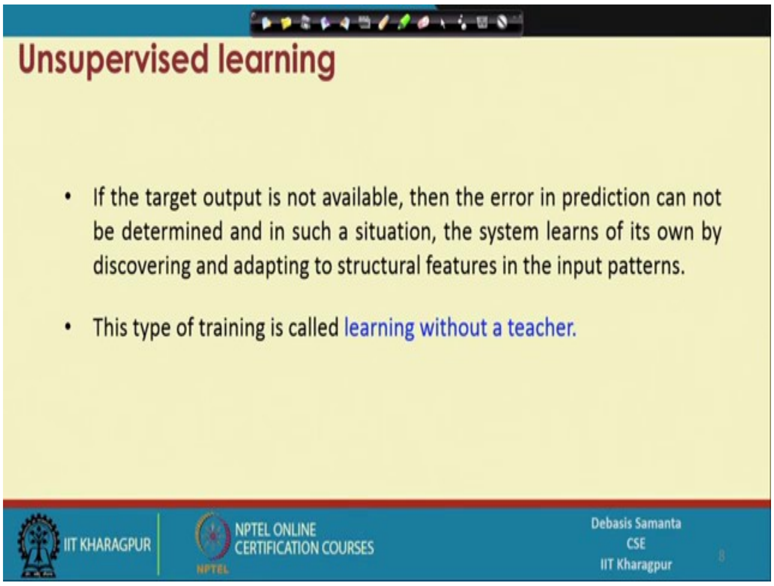
(Refer Slide Time: 07:50)



So, in the few next slides, let us quickly learn about what us the different learning techniques are there. I will first discuss about Supervised learning. Now in each learning, one thing is common that we need training data or a set of data that can be used to learn the network. So, training data is there. So, so in this basically, in case of supervised learning, so for input what will be the output it is already known to us. So, that is why this is called the supervised training data. That means, for every input what is the output we should say it.

So, one hand written character, the input is there, then we say that it is A like this one this. So, this is why this kind of learning is also called learning with a help of teacher. So, basically teacher said the question or tell the, I mean ask the question and then also if you are not able to give the answer, is also tell the answer also. So, this is why it is called learning with the help of teacher. So, Supervised learning is very simple and state forward in fact and yet is very effective also.

(Refer Slide Time: 08:55)

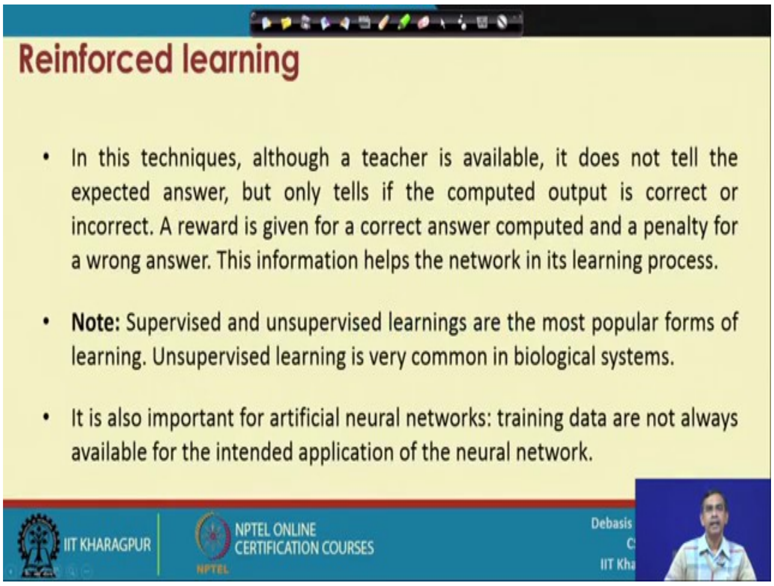


Now, the next is Unsupervised learning. If the supervised learning is learning with the teacher, unsupervised learning also can be termed learning without a teacher. So, here in this learning concept, if in case of supervised learning as I told you input is given then, output is there, but in case of unsupervised learning, input is only given; no output is told.

So, in this case for example, in case of character recognition, we can give 26 different patterns right that I write and 26 different patterns related to the different input characters and then the 10 if it is learned by means of unsupervised, then they can automatically decide about whether it is which character it is.

So, here basically output is not mentioned here, in the line of input; only the input is there. So, this is the Unsupervised learning concept is there.

(Refer Slide Time: 09:52)



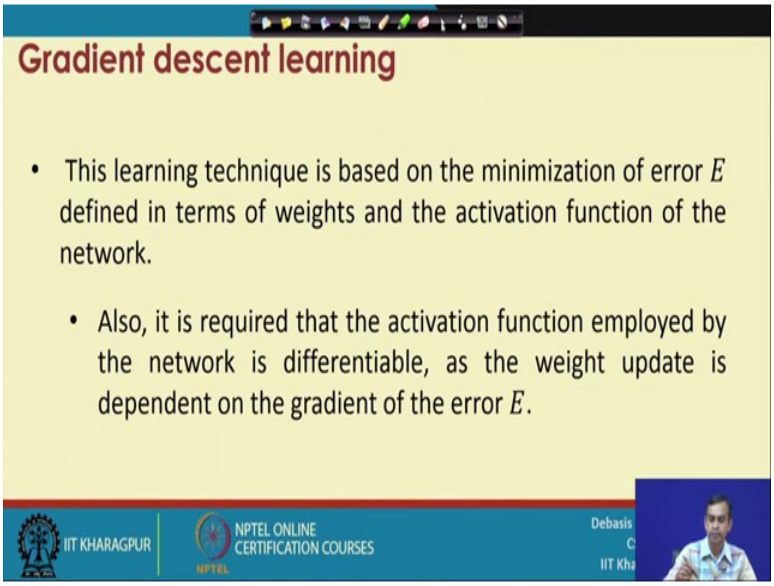
And the next is called the Reinforced learning it is the ok; supervised learning, unsupervised learning and the reinforce learning.

Now, in case of reinforced learning, a teacher is available like a supervised learning, but teacher does not tell the expected answer, but only tells that if the output is give answer is given, whether the answer is correct or incorrect. For a given correct answer, it basically reverses something and for the incorrect answer, it gives a penalty. Now knowing the questions and therefore, the reward or penalty they the network can learn it and this is the idea about Reinforced learning.

So, in the theory of learning, mostly the Supervised learning and Unsupervised learning are the most popular form of learning and Unsupervised learning is basically the common learning technique in our biological process. We people easily follow unsupervised learning. Otherwise, supervised learning is more things are there.

Now, so, here actually, so for the neural network learning or training is concerned, we heavily depend on the input data or the training data it is there. Now so, these are the learning techniques the 3 different learning techniques are there.

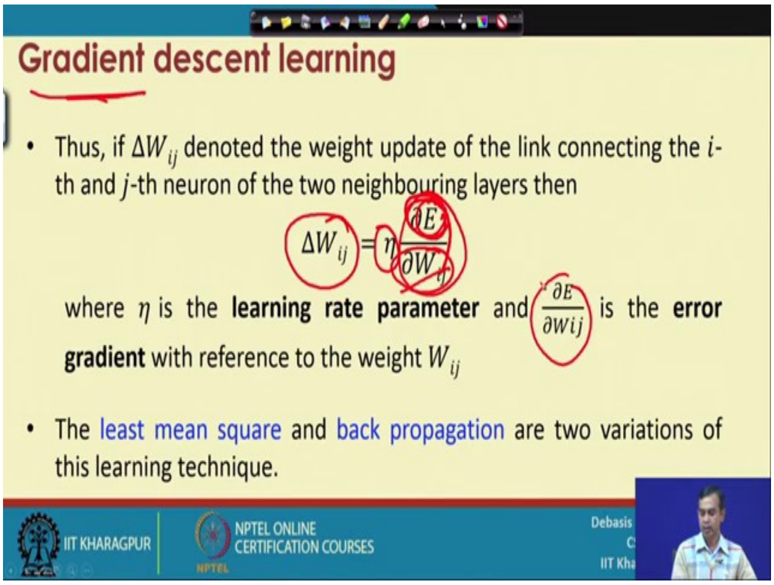
(Refer Slide Time: 11:16)



Now again, so far the supervised learning technique is concerned. There are again 2 types the gradient descent learning that can be by means of least mean square or it is basically the error descent, gradient descent calculation formula is there or back propagation is there. We will discuss these things in details so that then it is basically, it basically calculates the error it is the input and output is known. So, it basically calculates the error and it basically try to optimize the parameters in the network so that this error values can be minimized.

So, this is the main idea about this gradient descent learning techniques. It is basically supervised learning techniques in general.

(Refer Slide Time: 12:02)

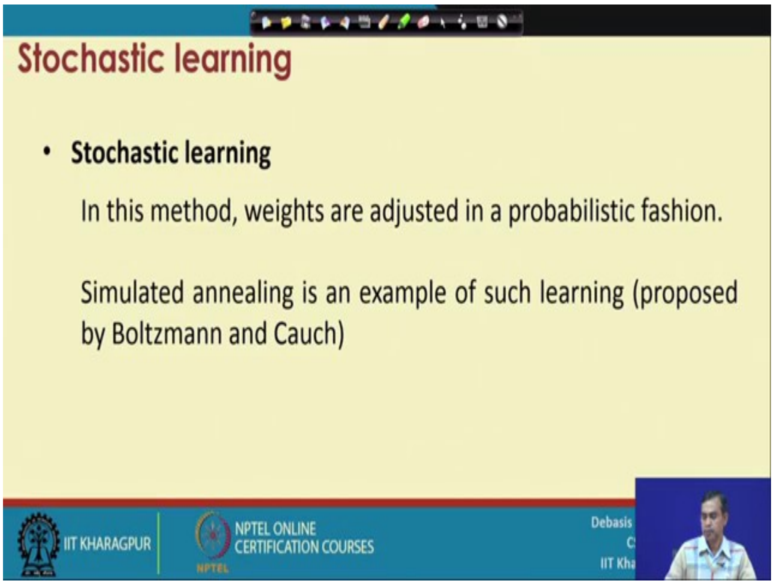


And, and the gradient descent techniques is basically, it is called the gradient descent as a gradient basically the gradient concept is there. So, gradient is because it basically, if E denotes the error at any instant or and then if we change the weight matrix for example, then the what will be the change of errors.

So, it will change this basically, it will calculate if the weight matrix E changes then it will see that how the error will be there and then accordingly the changed weight matrix will be calculated. And here η is the one constant, it is called the learning parameter. And this ∂E by ∂W_{ij} is called the error gradient.

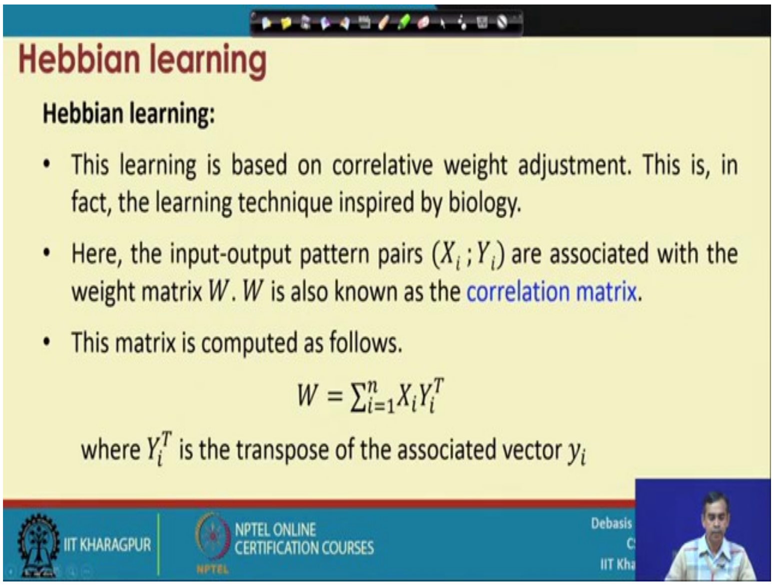
So, basically it will see exactly, if we change this weight then what will be its error and then accordingly for the minimum errors, it will decide the weight changes and that will be the learning parameters actually. So, this is the idea about it. Now whenever this error needs to be calculated. So, for this error calculation in this gradient decent method 2 technique; either least mean square error or back proportion error is there. So, we will discuss about back proportion algorithm and least mean square error calculation in the training methods.

(Refer Slide Time: 13:17)



Now, so the supervise learning also can be a Stochastic learning. In this method, the different network parameters like weights and others are decided or adjusted in a probabilistic fashion. So, with certain probabilistic or with certain uncertainty, it can be decided. An example for this kind of is simulated annealing. We could not discuss the simulated annealing because the timing constant. So, it is not covered in this course. So, the simulated annealing is one kind of stochastic learning which is used. It basically simulated annealing it used to solve optimization problem solving.

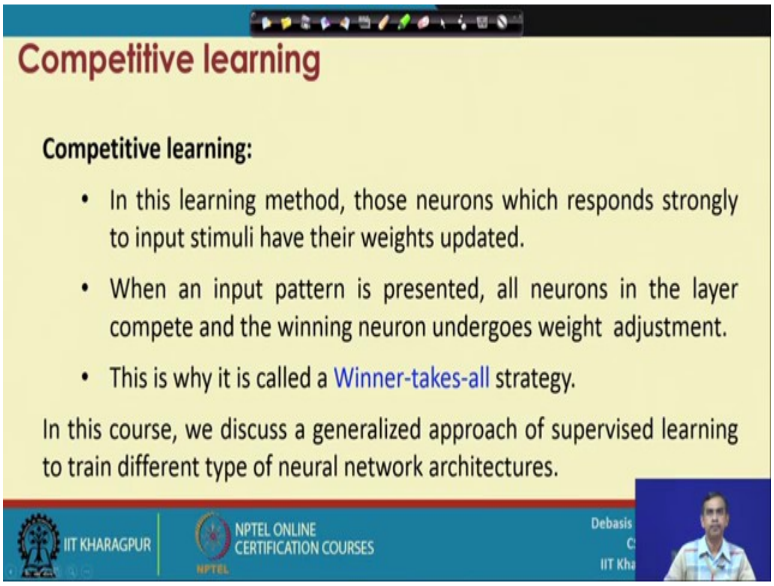
(Refer Slide Time: 13:56)



Now, next is so for the Unsupervised learning is concerned, there are 2 types of learning; one is Hebbian learning, another is Competitive learning. So, Hebbian learning is based on correlation analysis, is basically correlate weight adjustment. So, correlation on a statistical method which basically followed to decide if we decide this values, then how it is correlated with the actual output like this one. So, the correlation analysis and statistical things are involved here. So, a little bit mathematically complex, but it is also useful. So, for the heavier, so far the Unsupervised training if we want to adopt it.

Then it is called the Competitive learning.

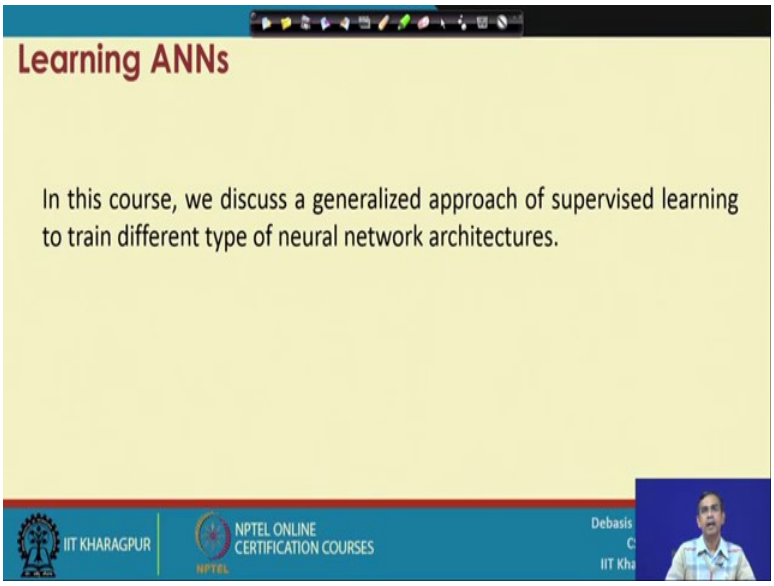
(Refer Slide Time: 14:43)



So, in case of competitive learning, so for certain input, if we see the neurons which responds very strongly; that means, it gives a lot of I mean different I mean input signals that it can passed or summation of the values are very strong, then it will decide these values as the learning parameter or this need neural network parameter for that particular neuron which basically responded to a particular input strongly is the parameter that is taken here.

So, it is called the competitive because in this case, the neuron which will basically responds in a strong manner is the winner. So, that is why this kind of learning, this kind of unsupervised learning technique is called Winner-takes-all strategy anyway. So, we have learned about briefly the different learning techniques and in this particular course, it is not possible to cover all the learning techniques. We will cover only the generalized approach of the supervised learning techniques and then we will also discuss about the different architecture to learn it.

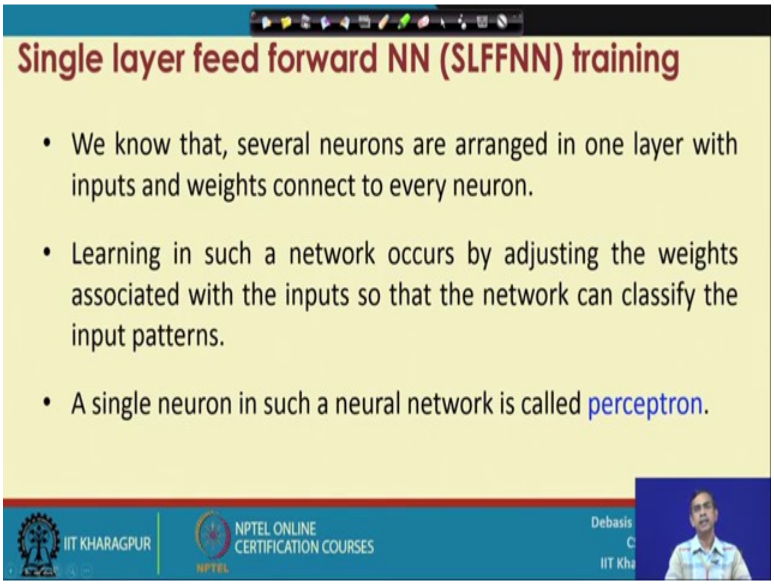
(Refer Slide Time: 15:55)



So now, so that is why we have limited our approach to generalize approach of supervised learning to different type of neural network architecture. Now let us start from first training concept with the help of the simple one, training methods to train the Single Layer Feed Forward Neural Network.

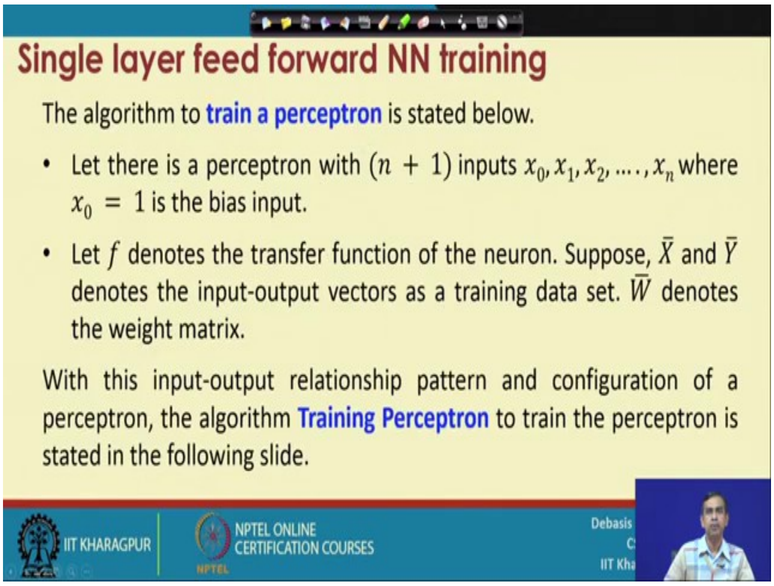
So, we will be learning, we will see exactly how a neural network can be trained and we will consider the approach, the supervised training approach.

(Refer Slide Time: 16:25)



So, let see how the single layer feed forward neural network can be trained and then the thing is using Supervised learning approach.

(Refer Slide Time: 16:35)



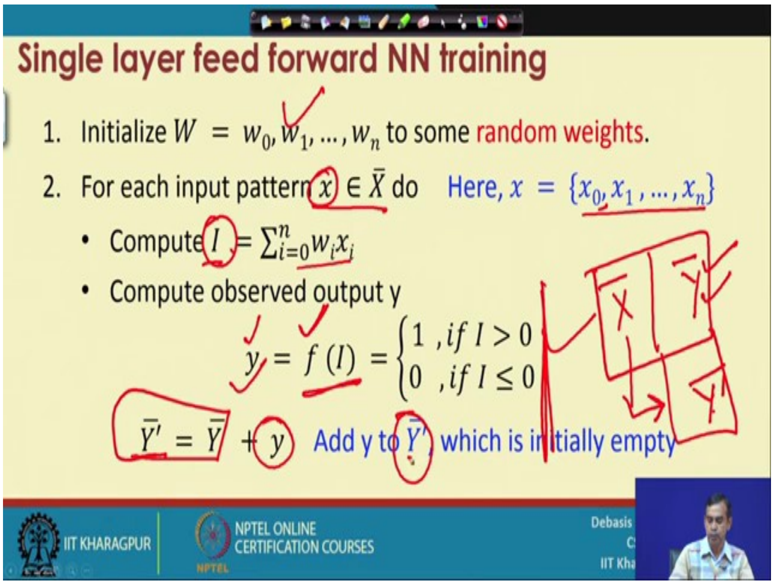
Now, so first we will discuss about training one unit, the basic unit. That means, training a perceptron. Now in order to training a perceptron, we will consider the inputs. Let us consider these are the input and one input is basically the threshold values input is a bias input also called. So, these are the actual input n number of input erudition the bias input. So, they are (n + 1) inputs are there.

So, we have to basically trained a perceptron and to train this perceptron, we consider that n inputs are there. So, this is basically the inputs to the to the perceptron. And for this perceptron, let us assume this f . f denotes the transfer function. Now we have discussed there are many transfer function like say step up the sigma transfer function, log sigmoid, 10 sigmoid and all these things are there.

So, it can be anyone. So, as we can decide any one transfer function for the perceptron; let it be the step up function. Then we will consider the supervised training as I told you. So, this $\acute{X}$ and $\acute{Y}$ denotes the input and output vectors as a training data set because it is the input data and this is the output data that is there and $\acute{W}$ denotes the weight matrix.

Now, with this input-output relationship pattern and configuration of a perceptron, we are going to discuss about one algorithm. It is called the training perceptron to train a perceptron which is ok. So, we will see that how it can be trained or it or a perceptron can learn from this combination ok.

(Refer Slide Time: 18:36)



So, let us see the learning perceptron algorithm. I am just going to define the different steps in it. It is very simple.

So, let us consider W the weight matrix in this case. So, this weight matrix is basically has the n+1 number weight value because n+1 inputs are there. Now initially all the values in this weight matrix are chosen with a random value. So, random weights, so weight matrix can be normalized in the scale of 0 to 1 and these are the values are there.

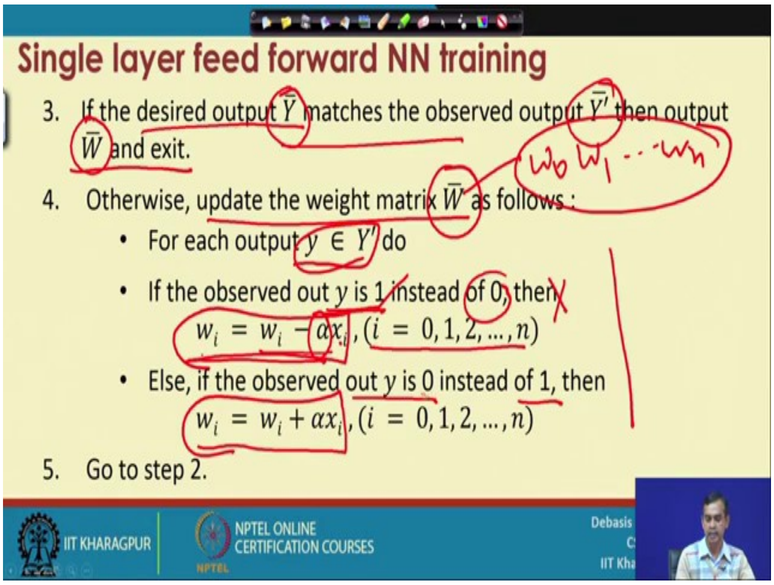
So, initially, this weight matrix is initialized with some random values. So, this is the initialization of the weight matrix and for each input pattern x belongs to this $\acute{X}$. $\acute{X}$ is a input data set where x is like this one. We have to compute I. I is basically using this w and this I we can calculate I. Then for this perceptron and with respect to a particular input x, we will be able to calculate y that is $f(I)$ where f is a another function that is assumed ok. As I told you it is a suppose a heavy side transfer function is assumed or sigmoid transfer function is assumed anyway.

So, take let us take with a. And then, so for this input I, we have to see if I > 0, it is a 1 and if I < 0, it is 0 this is basically the simple transfer function or it is called the step transformation that we have considered $f(I)$ are in this case.

Now, it will result 1 y. Now in the decide or observe output, we will store y' . We add this output into this y' . So, we have decided earlier $\acute{X}$ and $\acute{Y}$ are the input and output as the training data and here for each input whenever we given it to the perceptron, we see for that input what is the observed output. So, it is basically actual output and this is basically observed output. So, for each input here, we will be able to obtain the observed output using this method, this concept. So obviously, this observe output is a set of all outputs, initially it is empty.

Now, this is the concept here; now next let us proceed then.

(Refer Slide Time: 21:08)



Having these are the for each input, we will be able to calculate the observed output. Once the observe output is known to us, we will be able to match it. So, there is a matching. So, this is basically the actual or true output. If the true output matches the observed output $\acute{Y}$, then we can say that the perceptron is learned correctly. And the output in that case whatever the output $\acute{W}$ is basically the output of the model. So, model is ready.

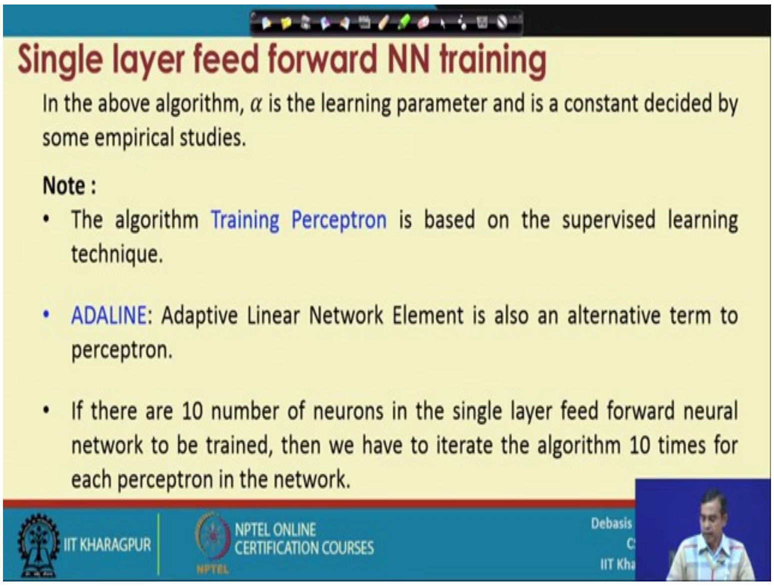
On the other hand, if it does not match correctly, then there will be the few conditions are there. We have to update the weight matrix $\acute{W}$ because then the network is not learning properly. So, in this case, what is the learning procedure is that. So, basically we have to change the weight matrix w_0, w_1, and w_n, weight matrix is to be to be changed or updated.

So, that updation can takes place in this algorithm like this for each output y in this observed output set; if the observed output y is 1 instead of 0 actually, then the we can adjust this w 1 output with this formula. Where alpha is a constant will be decided by the programmer. And for each I, 0 to 1, we have to calculate these values for each x_i. So, each weight will be adjusted if we see that output is 1, but it should be 0; instead of 0 it is coming 1. If the output is 1 and it is also coming 1, so no need to do these things. On the other hand, if some other output is 0 instead of 1, then we have to adjust a weight matrix in this formula. So, it is basically this formula when output is 1 instead of 0 and in this case output is 0 instead of 1.

So, or the weight value will be adjusted and for each input pattern that this is a new weight is there. Now with this new weight, again we can repeat the same thing what we have learned earlier; that means, this step is to be repeated. So, go to step earlier and then repeat again the same thing until we can find the full matches of this one. So, it is a repeative process, it is an iterative process. So, that the same input data can be used with the refine revised weight each time and then we can proceed until we can get it this one.

So, this way the, in this case the w parameters can be checked with one function say output function, it is the transfer function is called the step up function. The same algorithm again can be repeated using the different transfer functions and then again the same thing can be checked whether the output is coming properly or not. And the same the technique can be repeated to decide what will be the threshold function and so on. So, this way for each network parameter we can repeat it and then a perceptron can be trained. So, this is the idea about how a perceptron can be trained or how a perceptron can learn from the supervised data.

(Refer Slide Time: 24:24)



Now, this algorithm, the training perceptron based on the concept of Supervised learning technique and this is also called ADALINE because it is Adaptive Linear Network Element and basically we are in this case we have considered training or learning a single perceptron. Now our objective is basically to learn all the learn the entire what is called the network basically SLFF NN network the Single Layer Feed Forward Neural Network. So, in that case, if the single layer feed forward neural networks counter n number of inputs and n number of layer suppose, we can n number of neurons in the layer for example, say 10 neurons in the layer then we have to learn each neuron at a time and then that learning process is to be repeated 10 times for every for all the perceptrons.

So, learning is basically same approach, but needs to be processed, needs to be computed or each one, one by one. So, this is the idea it is followed there in case of Single Layer Feed Forward Neural Network architecture training ok.

So, this is the concept of Single Layer Feed Forward Neural Network architecture learning we have learnt. In the next class, we will learn about learning the multilayer feed forward neural networks and then we will consider about one technique that is required to learn a neural network technique is called the error gradient descent method using back propagation algorithm which will be discussed in the next few lecture classes.

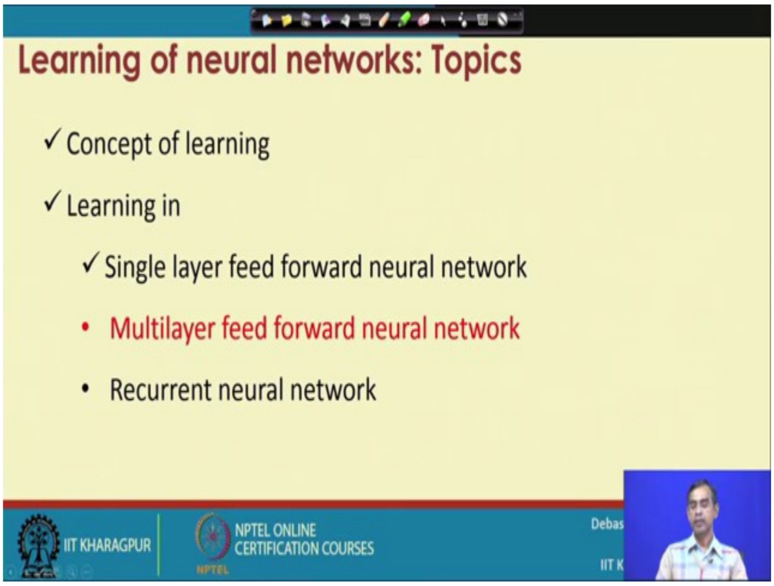
Thank you very much.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 37
Training ANNs (Contd.)

So, we are discussing Training Artificial Neural Networks.

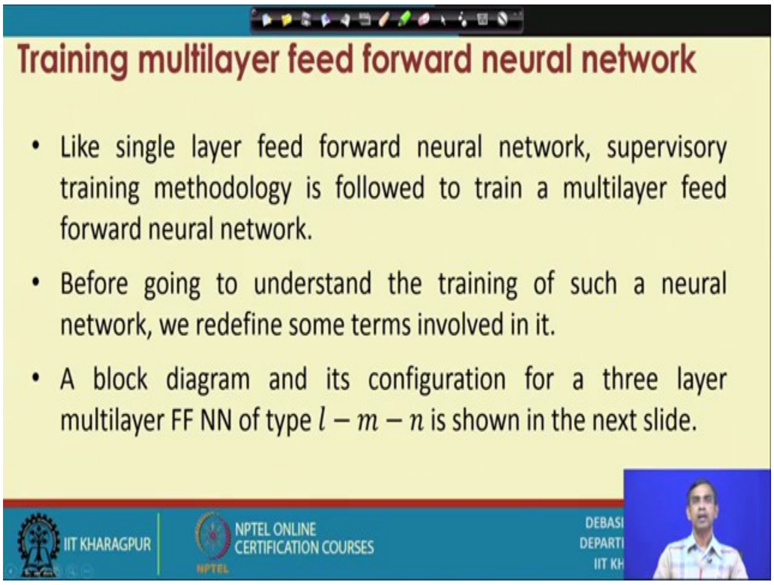
(Refer Slide Time: 00:27)



So, far we have discussed about so, how to train a neural net; it is basically based on the concept of learning. There are different learning techniques and we have considered the learning techniques again varies from different network architecture to different network architecture. In the last lectures, we have discussed about how a single layer feed forward neural network can be trained. Today, we are going to discuss the multilayer feed forward neural network training and then the recurrent neural network training will be discussed.

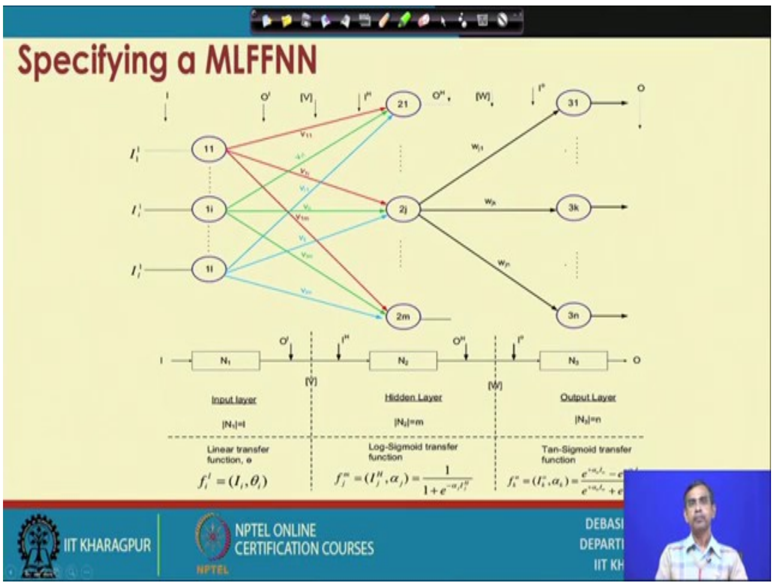
Now, so, multilayer feed forward neural network training, basically, the similar approach to that of the single layer, but it follows more method more what is called a meticulous method, particularly it is if you consider the supervised training then it considers algorithms to train it and the popular algorithm in this regard is called back propagation algorithm.

(Refer Slide Time: 01:35)



So, today will discuss about the algorithm the whole. Now, before going to discuss about training a multilayer feed forward neural network for the simplicity of the discussion we will consider a multilayer feed forward neural network with the configuration l-m-n that is called the l-m-n network and it is basically a three layer feed forward neural network and l basically the number of neutron in the number of neurons in the first layer and m denotes the number of neuron in the hidden layer and n is the number of neuron in the output layer or in other words l-m-n network is basically a network with l number of input and n number of output.

(Refer Slide Time: 02:28)



Now, let us see this figure, because our many discussions subsequently will refer from this figure only. In this figure we have depicted the architecture of a l-m-n feed forward neural network and in this architecture. As you see this basically the input layer and this is the hidden layer and this is the output layer. So, the input layer network is called the N_1 the network which is there in hidden layer is called the N_2 and hidden layer that is there in output is called the N_3.

So, we can see it is basically the cascading of 3 network N_1, N_2 and N_3 one by one is a cascading in the sense that the. So, this is the input to the input layer and here we can consider l number of inputs the l number of inputs are denoted as I_1, $I_{2,}$ … I_l and then subscript denotes that it is a input layer. So, so these are the input basically and so, the set of input we can denote it is I and these are the perceptrons in this input layer l number of perceptrons are there. So, they are termed as 1 1, 1 i, 1 l like this one. So, one indicate that it is a first layer and then next symbol increases that which make symbol if it is I indicate that is i-th perceptron in the first layer.

Now, the input to any perceptron is basically this one. For example, to this perceptron input is I_l to this perception input is this one and. So, on now output of a perceptron in the input layer we denote it is O^i; O^i denotes the output of the input layer and here we can say that the output of the input layer is basically input to the hidden layer. So, input to the hidden layer we denoted I^H. Similarly, the output of the hidden layer we denoted O^H. So, these are the output of the hidden layer and the output of the hidden layer is basically output of the output layer and input of the output layer we denoted I^O. So, these are the basically input of the output layer and finally, this is the output layer is called the O.

So, here basically I and then O input and output through this network right it basically mapped from an input to output. So, this is idea about it. Now, so, this is the working of the multilayer feed forward neural network more specifically l-m-n network now here will consider few more things as I told you this is a network n and size of the network N is l similarly, this is our network N_2 size of the network is m and this is a network N_3 size of the network is n. So, l-m-n like.

Now, in each perceptron here in the input layer they will follow each perceptron they will follow transfer function and then thresholding value for the simplicity for. So, this is the idea. So, if this contains the first perceptron in the input layer we denote the transfer function as f_i^1. So, f_i this is the f_i means the transfer function of the i-th perceptron in the input layer this one.

So, this is basically it basically I_i, I_i and Θ_i where Θ_i is the threshold value for this perceptron. So, what I want to say is that, so, each neurons are characterized with their own transfer function and then thresholding value and then input when give it pass predict through this input it will produce output. Now, this output then will be fitted to all the neurons in the hidden layer. So, this output goes to this layer this layer this perceptron and this perceptron and this one.

Now, whenever this output of the input layer goes to the goes does an input to the hidden layer they will be associated with weights. So, from the perceptron one in the input layer if it is goes to the perceptron one in the hidden layer we can say that it is weighted by V_{1l}. So, similarly we denote V_{ij}, that means, it is a weight for the input when it passes from the i-th perceptron in the input layer to the j-th perceptron in the hidden layer, so, V_{ij}. So, this basically constitutes the weights for all signals that can be fed to the hidden layer and we can represent this weight by means of a matrix, V matrix that we have already discussed and it is similar the that of the single layer feed forward network. So, in this case the matrix is size $l \times m$. So, this is basically l m matrix, l number of rows and m number of columns are there.

Next, the output of the hidden layer goes to the goes as an input to a where each perceptron in the output layer for example, if j-th new perceptron it is there in hidden layer so output from this perceptron goes there and goes all the perceptrons in the output layer. Now, likewise the input to the hidden layer here also all input to the output layer will be associated with weights. So, these are the weight matrix and this weight matrix is denoted by W_m and in this case the size of the weight matrix is $m \times n$, because from m number of perceptron it goes to the n number of perceptron in the output layer. So, $m \times n$.

So, it is a similar the V matrix that we have discussed in the previous in between input to hidden layer here also W matrix is between hidden layer to output layer. Now, again for each perceptron in hidden layer it will be characterized with transfer function and then thresholding value. So, if it is in the j-th layer j-th perceptron in this layer then we can represent that f_j^m and this basically the input to the j-th perceptron and this α is a threshold value. We can assume here that all the transfer function that it basically here for each perceptron is denoted by this one. It is basically log sigma transfer function and in this case we have considered a linear transfer function, that means, it is simply the pass input to the output.

So, so, these are the two transfer functions and threshold values are there in input layer and then output layer. Now, similarly the transfer function that we have considered here for any case perceptron in this layer is denoted by this one and here we have considered the log the Tan-sigmoid transfer function. So, for the sake of varieties we have considered different transfer functions in the different perceptrons with the different threshold values in each layer.

So, this completes the description of the element network as a multilayer feed forward neural network. So, now, once this architecture is clear then we will be able to see how this network architecture can be trained. So, trained means there are many things are to be learned. So, far training is concerned. The first of all how many numbers of perceptron should be there in the input layer that obviously, specified by the number of inputs of course, similarly number of perceptrons in the output layer it is also decided by the output.

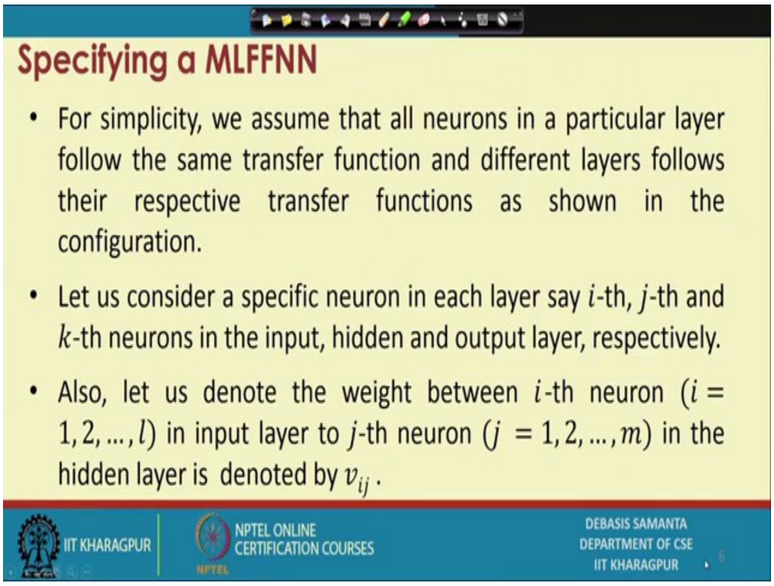
So, these two things are very simple, right it is, but so, for the learning is concerned how many neurons are there should be in the hidden layer it also needs to be learned. So, the n this m value needs to be a plan. So, this is an another learning parameter and then we have considered the threshold value in each perceptron in the input here. So, this values also needs to be learned, then α_j value for each perceptron also needs to be learned and then here also α values for each perceptron to be learned. So, these are the learning parameter therefore.

So, m to be learned Θ_i values for each perceptron α_j values for each perceptron α_k values for each perceptron are to be learned. Another also the things to be learned that we have considered here in our discussion that this is a transfer function, but it can be other transfer function always as well. So, there many transfer function. So, it is also needs to be learned which transfer function will be better so far the accurate output is concerned. So, all the transfer function that we have discussed assume that these are transfer function, but here also system should learn or the transfer function that it should be here. So, these are the different things to be learned.

So, these are the objective of learning. So, learning means we are fine I also forgot to mention one thing another learning parameter is V and W. So, these are most important metrics or parameters to be learn V, W. Now, in this discussion it is not possible to discuss all the learning parameters, but they can be learned in the same way the any one other parameter can be learned and for the simplicity in our discussion we will consider only how this network architecture should be trained. So, that it can learn the V matrix and W matrix for the application. So, we learn about how to how the network can be trained so that this network can learn V matrix and W matrix and following the same approach we can learn other network parameters.

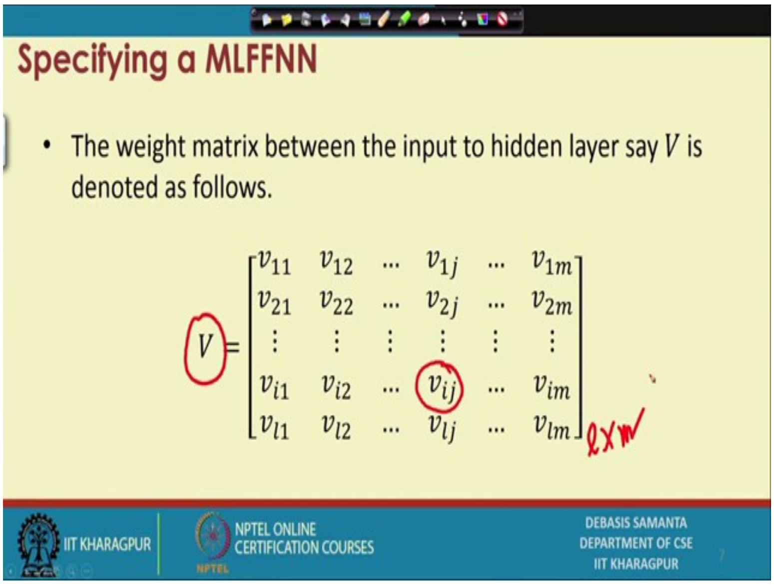
So, so, this is the objective of the learning that we are going to discuss.

(Refer Slide Time: 13:12)



Now, ok fine and for the sake of discussion we will consider these are notation that we will follow so that we can understand the things completely. We can say any neuron in the input layer as the i-th neuron i is 1 to l . Similarly, any neuron in the hidden layer we denote it as j-th neuron and any neuron in the output layer we denote in the k-th neuron.

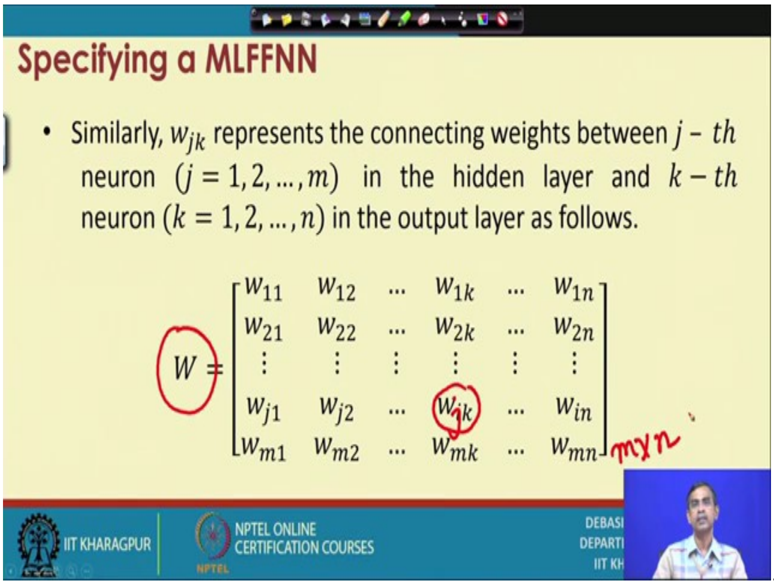
(Refer Slide Time: 13:43)



And, similarly there is a weight associated from i-th neuron in the input layer to the j-th neuron in the hidden layer, we denoted by a V matrix and the V matrix is like this. So, this is the usual V matrix that we have already used it in the single layer feed forward neural network training.

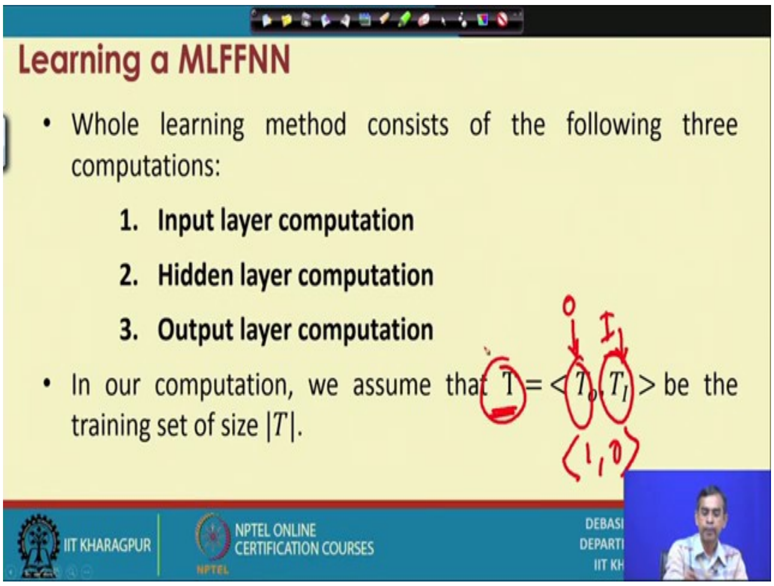
So, it is basically V_{ij} denote the weights from the i-th neuron to the i-th neuron in the output layer to the j-th neuron in the hidden layer and their values of i will vary from 1 to l and values of j will vary from 1 to m and this way it is an $l \times m$ matrix. Now, so, this is the matrix basically we have to learn it by means of training we should learn the different elements here in this matrix.

(Refer Slide Time: 13:43)



Now, likewise we denote the w_{jk} representing it is a weight from the j-th neuron to the j-th neuron in the hidden layer to the k-th neuron in the output layer. Now, all the weights that is there in this network can be represented by a matrix this is called the W matrix and W_{jk}, jk represents basically weight from the j-th neuron to the k-th neuron and it is therefore, a matrix of $m \times n$ size where m is a number of neuron in the hidden layer and n is the number of neuron in the output layer. So, this matrix W also needs to be learned and by means of training process we have to learn this matrix.

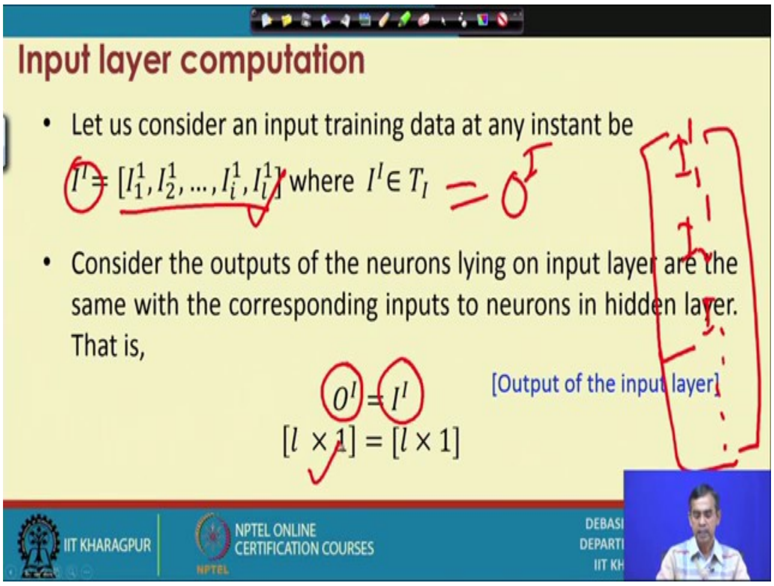
(Refer Slide Time: 15:25)



Now, so far the training is concerned basically training here in the neural architecture needs a number of computation. We can systemize the computational in the three steps method; first we have to compute the input layer, then hidden layer and output layer. So, we say that input layer computation, hidden layer computation and output layer computation. Now, so, far in our computation is concerned competition is based on some training set. So, here we denote the training set as T and T the training set consists of the input data and then output data for any input I which is belong to T_I has an associated O, <I, O>. So, it is basically I and O one sets, this sets is basically supervised training sets.

So, T is the training data here. So, given the training data right, we have to learn the different network parameter in this case V and W matrix. Now, to learn it or as a process of learning we have to learn we have to compute the different layers in each in the network.

(Refer Slide Time: 16:43)



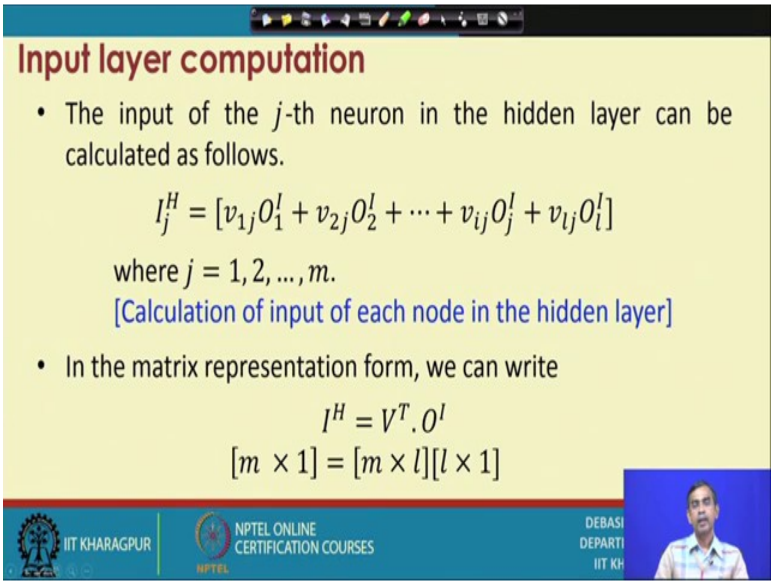
And, so now, let us see first the input layer computation. Now, the input layer computation can be discussed like this say suppose any input I^l which is in T_I, T_I you have discussed about is the input set and this a I consists of l number of inputs for each neuron to the input layer. So, we can denote I_1^I, I_2^I, I_3^I this one, ok. So, this basically is a one input that is there in the input set.

So, this is basically the one input which is belongs to this training set T_I and the output layer combination is prevail in this case this is because whatever because we have we have considered the linear transformation. So, linear transformation means y = x. So, so the whatever the value of x will be directly pass to the y. So, in this is why the output layer combination is that if O^I is basically the output instance at any time then give then the it is input is I_i^I or in other words if I_i^I that mean if this is the input then it is output is also O I it same because of linear transformation.

Now, all these things we can represent in terms of a matrix. So, all these right it can represent in a matrix like I_1^I, I_2^I and so on, so on. So, this is basically the matrix and as these l number of inputs data are there so, this is basically $l \times n$ matrix; similarly, I_1 also $l \times n$ matrix. So, what you can say that so, this is the input layer combination; that means, input to any perceptron in the input layer will be the output of that perceptron in that layer and this the any neurons input and corresponding output can be represented by means of this matrix formulation. This matrix of size $l \times n$, $l \times l$ and this matrix of size $l \times l$.

So, this is the idea about the input layer computation.

(Refer Slide Time: 19:07)



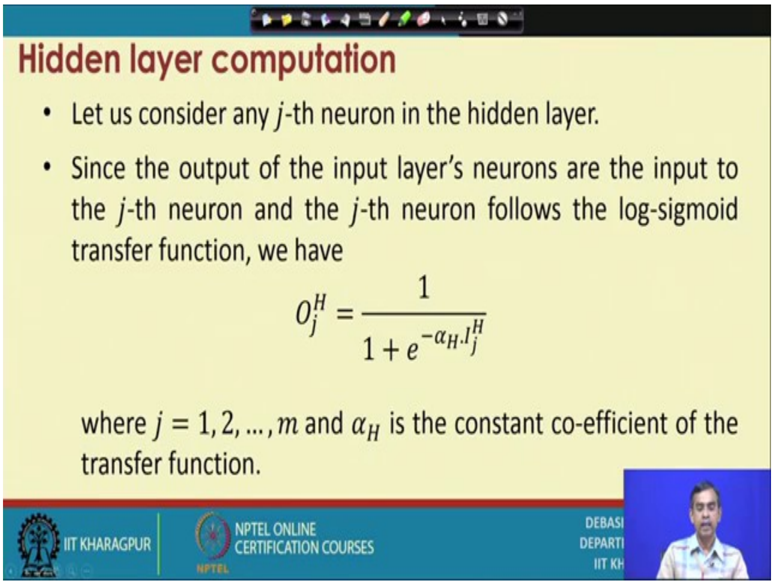
And, similarly we can say that if this is the output of a perceptron in the input layer then that will work as an input to the hidden layer. Now, if we consider any j-th neuron in the hidden layer then it is input can be considered as I_j^H. Now, this I_j^H can be expressed by this of product. So, basically it is from the first neuron to the j-th this is the symmetric the what is called the weight values and then it basically the output of the first neuron in the input layer.

Similarly, the second neuron to the j-th and then output from the second neuron in the input layer and this is a v_{ij} the weight from the i-th to j-th and it is basically the output of the j-th neuron in the input layer. So, this concept is there and this basically is a summation of all the inputs with their weights it gives the input to any perceptron in the hidden layer. So, I_j^H is basically input to the j-th perceptron in the hidden layer and this basically the calculation of input to the hidden layer, ok.

And, now this expression this expression can be represented in the matrix form which is represented here. Here it basically I^H is basically all the inputs to the hidden layer it can be represented V transpose matrix and O^I matrix. So, O^I basically output of the input layer and V transpose matrix is a V matrix and it is transpose form. So, in other word, if I^H is basically m cross 1 matrix and V-th is the transpose V is a $l \times m$ and it transpose from m cross l matrix and O_l^I is a l cross matrix. So, whole the things or whole this input layer computation can be expressed, in the form of a matrix representation. And, what we are observing is that all the calculations for example, input to the input layer or output of the input layer and input to the hidden layer is basically input layer calculation and all this computation can be expressed in the form of a matrix as it is here.

So, this is basically matrix representation and then matrix operation and then neural network training and then it is describing the model is nothing, but a matrix model or matrix formulation of the model.

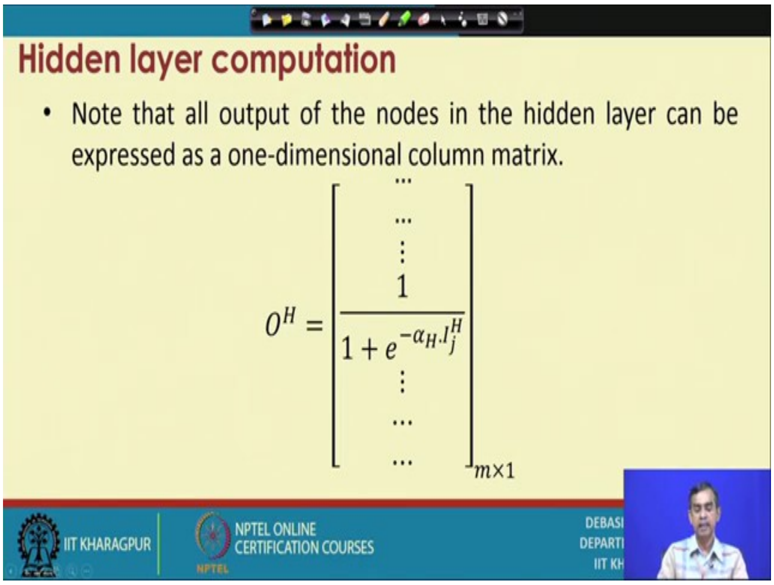
(Refer Slide Time: 21:48)



So, this give the input layer combination and then we will come to the hidden layer combination. Now, we know exactly what is the input to any perceptron any j-th perceptron in the hidden layer. Now, our task is to calculate what is the output of the j-th neuron in the hidden layer. We represent the output of j-th neuron hidden layer as this form O_j^H and as you have discussed that the transfer function that it used is basically log sigmoid and it has this form and these are the alpha H indicates that it is basically it is α_j better we can write that it is basically the threshold value of the j-th neuron in the hidden layer and I_j^H if it is a input. So, it is basically e to the minus alpha I concept. So, it is the output. So, this output is basically for the j-th neuron in the hidden layer.

Now, the way we have expressed the matrix representation. So, this calculation or the output combination also can be represented in the form of matrix and the matrix will look like this.

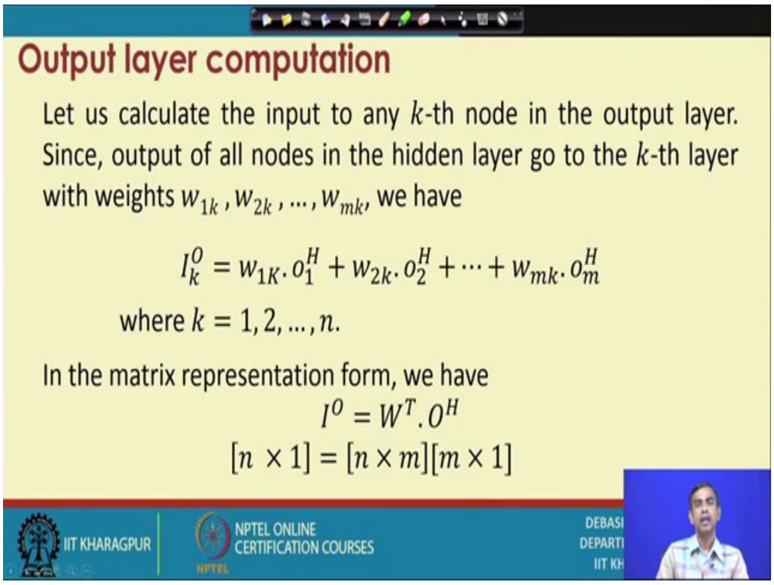
(Refer Slide Time: 22:55)



It is basically that matrix we can say that O^H matrix is the O^H matrix. This means the output of the hidden layer and output of the hidden layer because m number of perceptrons are there. So, it is a matrix of size $m \times n$. So, it basically includes the output for the first perceptron, second perceptron and these the m-th perceptron. So, it basically matrix that mean all the outputs of the perceptrons in the hidden layer can be represented by this type of matrix. So, this is another matrix to compute the hidden layer.

Now, so, the hidden layer output is known, hidden layer input is known, input layer input is known, input layer output is also known.

(Refer Slide Time: 23:45)



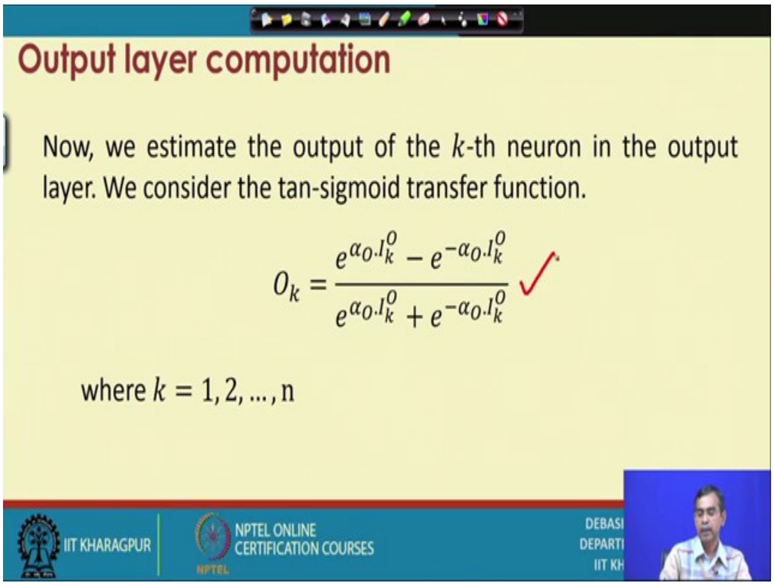
Now, we are in a position to discuss about output layer computation. Now, in case of output layer, we know output of the hidden layer works as an input to the output layer. Now, if there is any neutron say k-th neutron and we denote the I_k^O represents that it is basically input to the k-th neuron at the output layer. So, is a input to the k-th neuron at the output layer.

Now, this input is basically is a summation is a sum of the products of the weight matrix corresponding to the output from all the perceptrons from the hidden layer. So, this can be expressed this is basically output from the first layer in the hidden layer, output from the second perceptron in the hidden layer and output from the m-th perceptron in the hidden layer and is multiplied by this product is the weight matrix, weight values from the first neuron to the k-th neuron from the second neuron in the second layer to the k-th neuron in the output here and so on, so on.

So, this way this basically is an expression is the input to the k-th neuron at the output layer and there are n number of neuron. So, k will vary from 1 to n. So, this basically computes the input to the output layer. Now, this expression can be expressed in the in terms of matrix representation which is here. So, I^O denotes all the inputs at the output layer and this can be expressed the transposition of weight matrix and multiplied by the O^H, O is basically output of the hidden layer, this output of the hidden layer we have already learn how to get it. So, this way the matrix representation of the output layer, input of input to the output layer can be obtained and we can represent it this matrix as this is a $n \times l$ matrix, this is $n \times m$ because W is a $n \times m$ matrix and this is an $m \times l$ matrix.

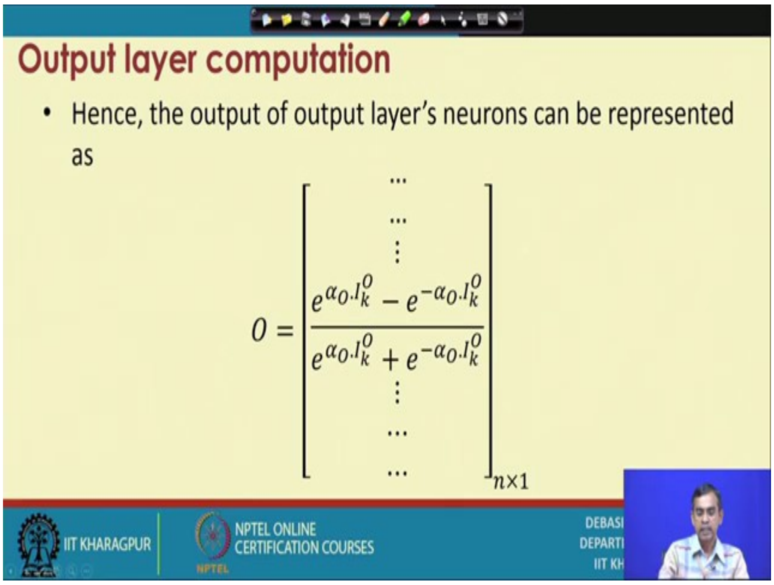
So, so, this basically the input computation at the output layer or is called the output layer computation. Now, we will discuss about the output of the output layer.

(Refer Slide Time: 26:03)



Now, the similarly the way the output of the input layer hidden layer we have calculated in the same way we can express the output of the output layer. Now, we can see that we have already mentioned that the transfer function that we have used they are in output layer is basically they Tan-sigmoid transfer function which take this form. So, it basically the output of any k-th neuron in the output layer. So, this is the output of any k-th neuron in the output layer and there are n number of neuron in the output layer. So, k values vary from 1 to n.

(Refer Slide Time: 26:48)



Now, so, this expression is for a particular perceptron. Now, as a whole for the entire network at the output side also can be expressed and that too can be expressed using the matrix representation. So, this is a matrix representation for the output layer. Output layer, here O denotes the all outputs that can be obtained from the output layer and this basically this is the entry of the output of the first neuron in the output layer and is a second neuron in the output layer and this is the nth neuron in the output layer.

So, these things also can be represented by means of a matrix and here α_O is denotes that what is the threshold values. Now, here if we can for the simplicity we can consider that only one values of α. So, α_I is basically the threshold value in the input layer for all the perceptron, for the simplicity we assume it. Similarly, α_H we do not the threshold values of all perceptrons in the hidden layer and α_O denotes the threshold values of all perceptrons in the output layer.

Now, this is a just simply an assumption that we considered same values of threshold values for all neutrons neurons belongs to a particular layer, but in actual practice it is not necessarily the same we can consider the different threshold values for different perceptrons in the network. But it will increase the complexity it will demand more calculation so, otherwise it is not the issue it is only the issue of computation time.

Now, we have discussed about the different layer computation and all these layer computations will be used to train the network because this is the ok, is a mathematical manipulation that how we can represent a neural network mathematically. Now, we have just now discussed that how the entire neural network can be represented mathematically and that mathematical representation in the form of a matrix representation. Now, in the next lecture will discuss about using these are the different calculation how we can train the network.

Now, for the training in multilayer feed forward neural network there are many training procedures, but in this lesson will discuss about a particular which is most popular the back-propagation algorithm. So, that we will be discuss in the next lecture slides.

Thank you.

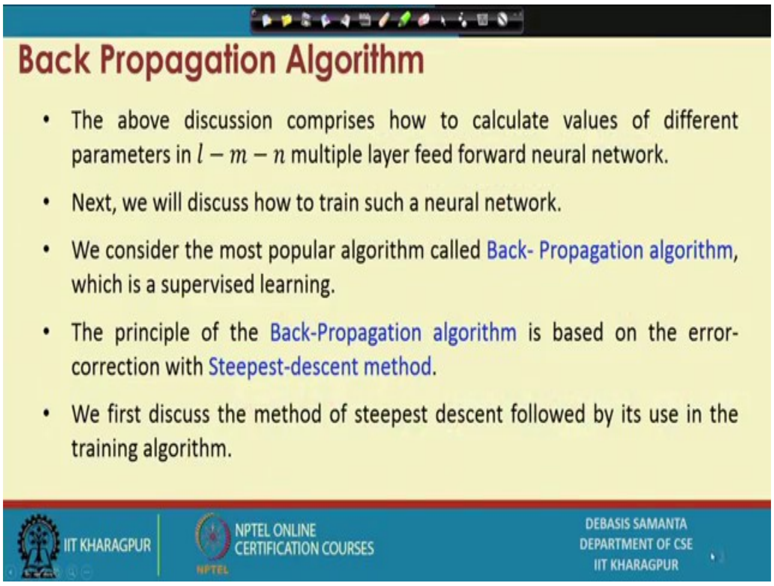
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 38
Training ANNs (Contd.) *Type equation here .*

So, we have model neural network, more precisely multilayer feed forward neural network and is a simplistic version of the multilayer feed forward network is element, where l is the number of neuron in the input layer, m is a number of neuron in the hidden layer, and n is the number of neuron in the output layer. Now, after modeling the element network and we have learned about that how such a network can be model and that model can be represented in the form of a matrix.

Now, we will discuss about once this network is model then how it can learn the different values that is there in the model. So, in particular we are going to learn about V and W matrix that is there in the model.

(Refer Slide Time: 01:13)

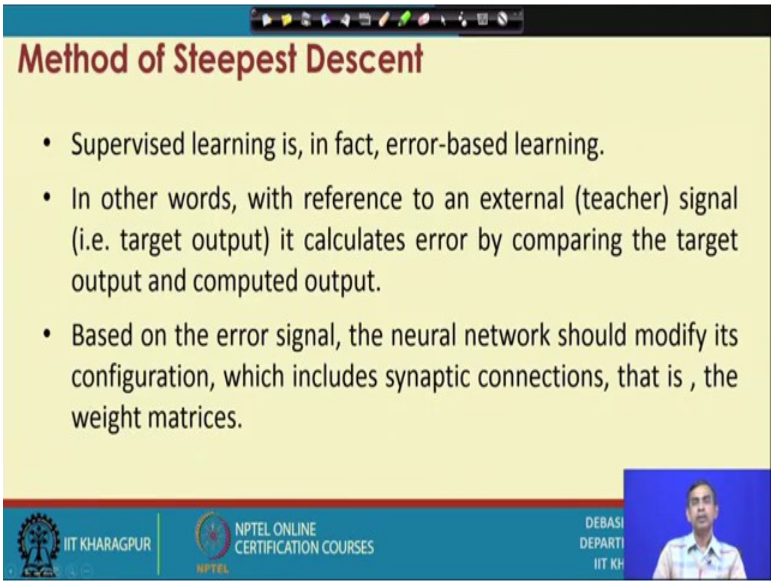


Now, so, the training algorithm that we are going to discuss is called the back propagation algorithm. So, it is a most popular neural network training algorithm and this algorithm is based on the concept of supervised learning. In this algorithm the basic concept that it is followed is basically error correction. That means, whenever some input is given to the network it will produce an output. So, it is called the observed output and as a supervised learning we know for each input what will be the true output. So, error is basically the difference between the true output and then observed output.

So, this propagation algorithm, back propagation algorithm tries to correct the errors; that means, it will train the network in such a way that the error that it will be obtained for a given input is as minimum as possible. Now, so, it is basically error minimization technique and error minimization technique which is followed here in back propagation algorithm or there are many error minimization techniques of course, like say a least square mean least square error method, but here we will discuss about one error minimization method it is called the Steepest-descent method.

Now, here I one thing you can notice that this back propagation algorithm is nothing, but finding the values of the different neural network parameters which basically minimize the error. So, it is basically an optimization problem. That means, we have to the objective function is to minimize the error value. That means, for a given set of input it will find the neural network parameters like V, W, l, m, n, Θ, transfer function and everything so that the error is minimum. So, this is the optimization problem in fact, so, back propagation is although also that is why it is called an, optimization problem it, right.

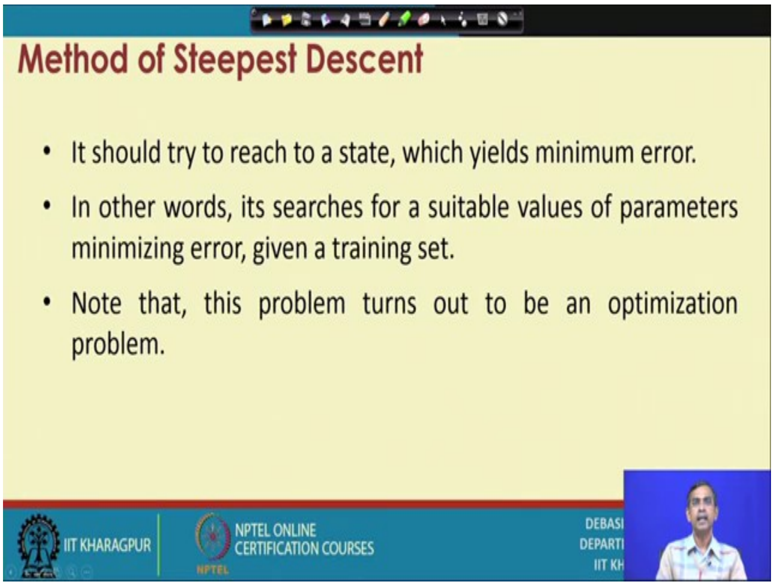
(Refer Slide Time: 03:28)



Now, first before going to discuss about this back propagation algorithm we have to learn about the what is the Steepest descent method and see this supervised learning is always error based learning and so, there is an error I already told you the error is basically difference between observed output and then the true output that is also called a target output and then computed output. Target output means it is a true output and then the resultant output the output which obtained from a given network is called the computed output.

Now, it will sense what is the error magnitude. So, based on the error magnitude the neural network should modify its parameters, its configurations. So, that is the concept that is followed. Now, again for the simplicity of the discussion we will not consider all the neural network parameters to be calculated. We will only consider the calculation of V and W matrix as the neural network parameter.

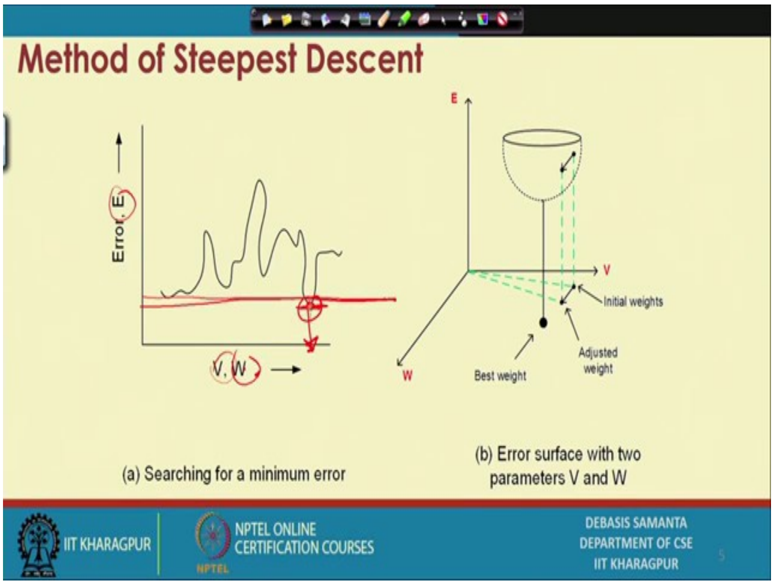
(Refer Slide Time: 04:42)



So, now let us see what is the Steepest descent method, that is there. So, basically idea it is that for the input it will produce the output and then thereby the error will be computed and then this error is basically a function of the neural network parameter values.

So, we have to set the values or we have to search for the right value of the neural network parameter so that the error that can be obtained is minimum. So, this is the concept that is there.

(Refer Slide Time: 05:12)



Now, let us see what is the Steepest descent method; it is basically here. So, as it is an optimization problem and if we consider V and W are the input parameter. So, for the different values of V and W we can have the different errors. So, this can be represented by this kind of variation that how the error matrix is varies with the different values of V and W.

Now, so far the minimum error is concerned so, this is basically the global minima. So, we have to find this is the V, W value so that this gives the minimum error. So, basically we have to search the V and W value over the entire search space and you have to find this point. So, that this V, W. It is just like a genetic algorithm concept also. So, that means, for the different values of V and W we have to find one V and W value. So, that the objective function here E is minimum. Now, this concept is followed here, but it is a little bit in a vector representation that Steepest descent is basically considered the vector representation and a concept is like this.

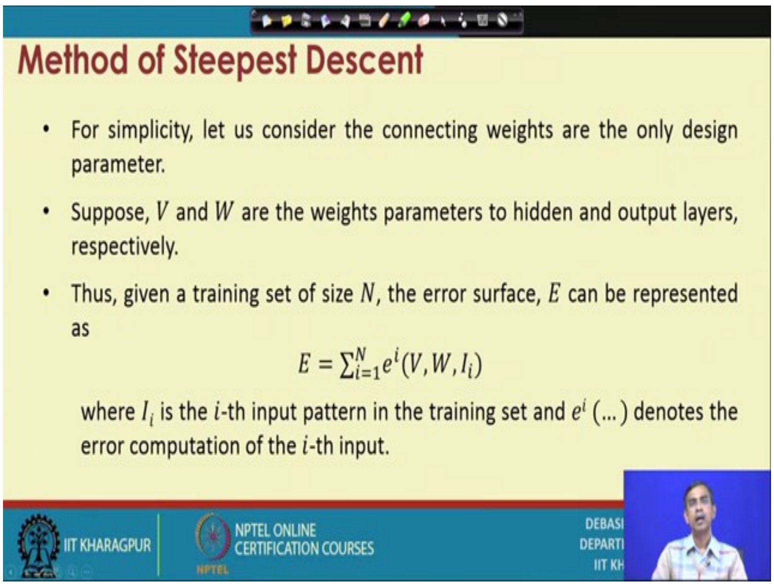
So, suppose at present this is the neural network parameter then from these values we can go anywhere in this any direction, but if we come to this value then this is basically called an error. So, it is basically how the error varies with the different V and W suppose, it is represented by this formula. So, so it is basically error versus V, W. So, it is basically the VW space and then error is there and suppose the error is represented by this. Now, so, at any instant if the neural network parameters are presented by this point then we have to find the next what is called a modification, so that this modification will go towards this minimum value.

But, it cannot go at one step from this to this one rather it is an is small incremental steps. So, from this V and W value it will come to this V' and W' values and then from another $V'W'$ value then ultimately it will V. So, this is basically incremental step and in each increments it will basically leads towards the minimum plato or it is called a minimum values of the objective function. Now, so, this way so this is basically the value of correct V, W value if at any present this one we have to search these values V, W out of this entire search space.

So, this is the problem and steepest descent method directs the network that if at any instant this is the V, W value then what will be the input what will be the values $V'W'$ at the next incremental level. So, steepest descent method tell about this idea and it is followed here. So, if this is the current weights then it is the adjusted weight at the next step. Now, we will see that how from the current weight and adjusted weight can be obtained; that means, from the current weights V and W, how the adjusted weight $V'W'$ can be obtained so that it will towards the minimum error.

So, this is the concept that is there in the Steepest descent method and we will discuss about the method the steepest how this Steepest descent method the concept can be implemented and then how this back propagation algorithm it is.

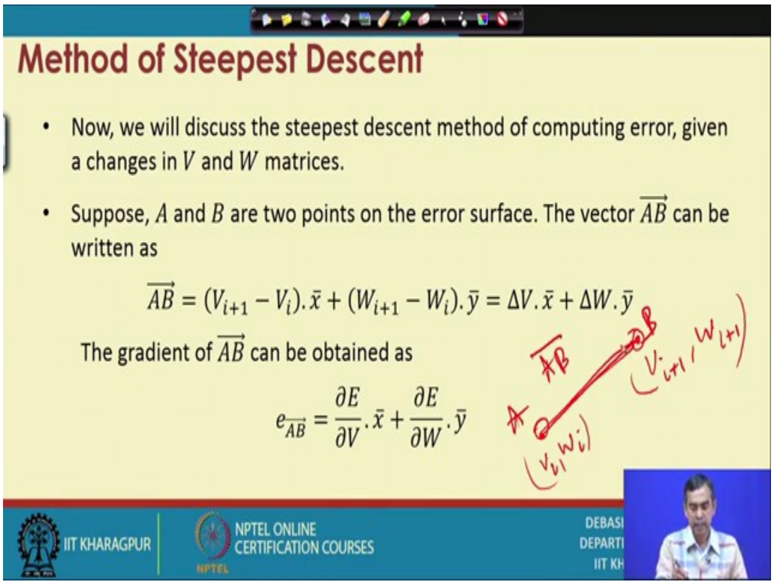
(Refer Slide Time: 08:35)



So, V and W are the network parameters. Therefore, the network parameters here V is basically the weight values to from the input layer to the hidden layer and W is the from hidden layer to output layer, right. So, in our in our context in this context we can represent the error function E. So, error function E is basically is a function of a three parameters the first parameter V and W because this error depends on the values of V and W and also the error depends on the input the input to the network and if e^i denotes the input due to the i-th if e^i denotes the error due to the i-th input to the system then the total error E can be expressed summation of all the errors due to the all inputs.

So, here N is the size of the training data if it is there then it is basically summation of all the N number of inputs that is there in the training set. So, this way the error function E can be described. So, error function E is therefore is a function of V, W and the input. So, this is the idea about representing error and then once the error is represented in terms of V, W and I_i we will be able to calculate error function and this error function E are to be minimized for a certain values of V and W.

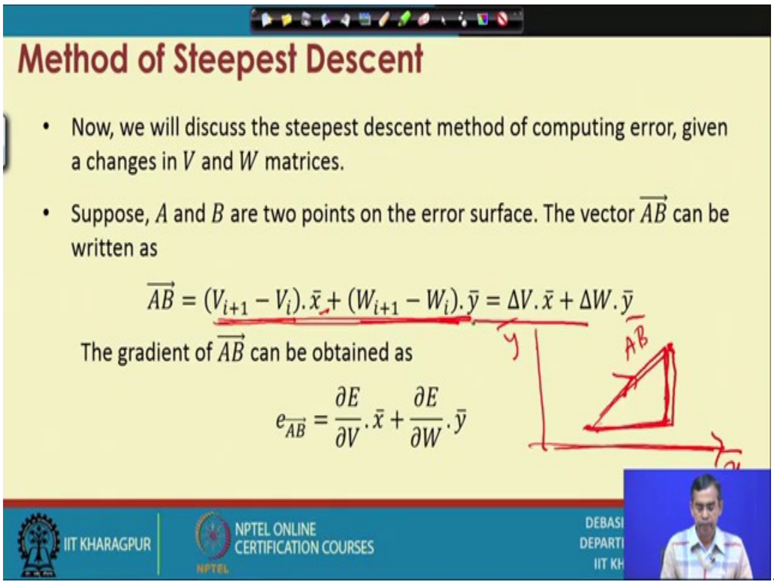
(Refer Slide Time: 10:33)



Now, we will discuss about the Steepest descent method how it basically search the right values of the neural parameters in the than nn parameters so far the minimum error is concerned.

Now, it required little bit the vector concept. Now, here suppose A and B are the two points. So, A is suppose V_i and W_i it is in the two dimensional space of the V and W and this is V_{i+1} and W_{i+1} are the next point, so, two points. Now, if the two points are given then we can represent this vector. So, here suppose it is A and it is B then the vector AB can be represented like this one. So, so this vector representation also can be represented in a vector form like this one.

(Refer Slide Time: 11:33)

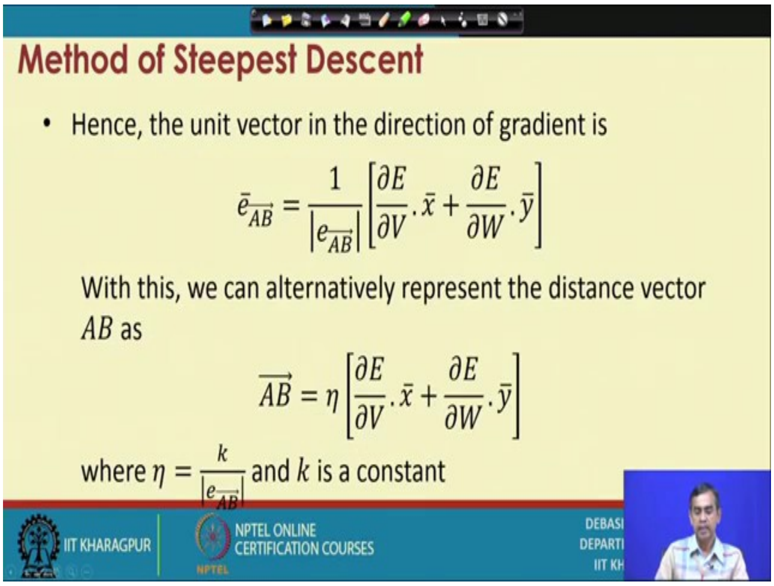


So, it is basically V_{i+1} it is basically this point and this is basically this is $\vec{AB}$ and this is V_i, W_i and $V_i + W_i$. So, this + this is basically representing this. So, here V_{i+1} - V_1 this one into x it is if it is x direction x and then it is W_{i+1} - W_1 this one so, it is this one and if it is a y direction. So, it is basically x and y if the two dimensional space then any vector in the two dimensional space having the coordinate axis x and y can be represented this form. So, is basically the representation of a vector with in terms of true points V_i and V_i W_i and V_i + W_1 in the search space and this can also be in a compact way can be represented this form where ΔV is same as this one and ΔW is same as this one.

Now, So, this is a vector representation if the vector representation is $\vec{AB}$, then we will be able to find the gradient; gradient is basically called the slope, the slope of a $\vec{AB}$ we can represented by this $e_{\vec{AB}}$ it is basically the gradient formula $\frac{\partial E}{\partial V}x+\frac{\partial E}{\partial W}y$, it is basically the error gradient.

So, it is basically if $\vec{AB}$ is the any vector then it is gradients will be represented by this form. Now, let us see how this concept is basically useful in the Gradient descent method here.

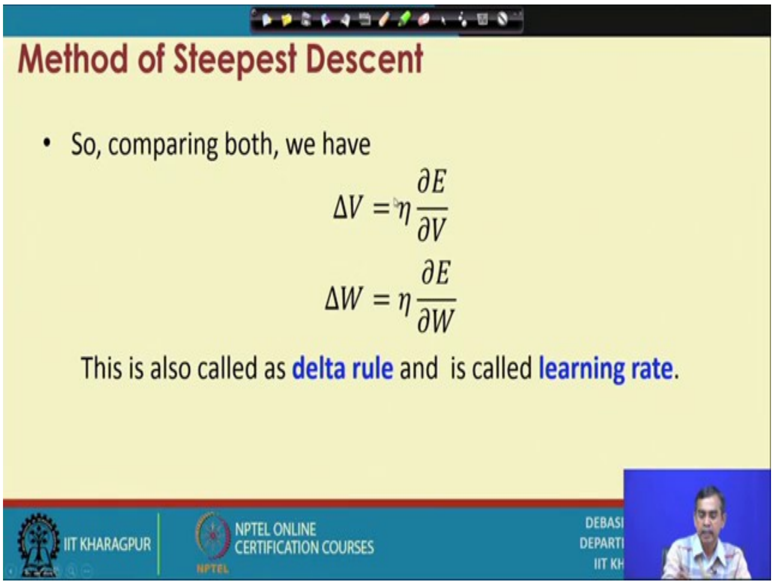
(Refer Slide Time: 13:11)



Now, if $e_{\vec{AB}}$ denotes the error gradient and then unit vector; that means, the unit, that means, the unit, that means, a unit vector for that a way a gradient can be represented by dividing the magnitude of the vector. So, it is basically the unit vector representation of the gradient vector. Now, so, and then if it is a unit vector it is the unit vector this unit vector can be multiplied by any scalar quantity give the vector itself.

So, every vector which we have given a representation in terms of V_i ,W_i and V_{i+1}, W_{i+1} also can be represented in terms of his gradient representation, where eta is basically the constant which is like this and these are the gradient formula that we have discussed about it.

(Refer Slide Time: 14:20)



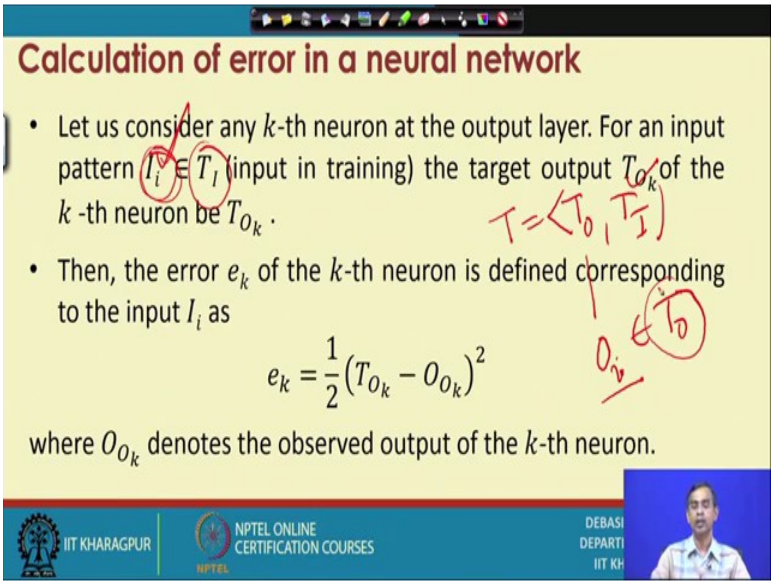
Now, so, having this is a representation and considering $\vec{AB}$ that we have discussing the previous vector representation and comparing this one we can readily write about this calculation as ΔV equals to this ΔV equals to ΔV equals to this one and ΔW equals to this one.

Now, this is a very important one observation of formulation. Particularly, this rule that we have just now obtained ΔV means what will be the change in V vector and ΔW means what will be the change in the W points. So, V and W are the two points then the changed either increase or decrease whatever it is if we represent it by ΔV then ΔV can be represent if E the error is known to there and then it is basically as a is $\eta \frac{\partial E}{\partial V}$. So, η is a one constant, this constant is called the learning rate.

So, here η is basically learning rate and this constant will be decided by the programmer. If the η value is not chosen properly then network can learn incorrect way and incompletely. So, the eta value needs to be chosen very properly and it is the programmers responsibility.

Now, we have learned about delta rule and delta rule is basically the Steepest descent method concept. Now, we will see exactly how the back proportion algorithm follows this delta rule to calculate this ΔV and ΔW value, that means, the network parameters.

(Refer Slide Time: 15:58)

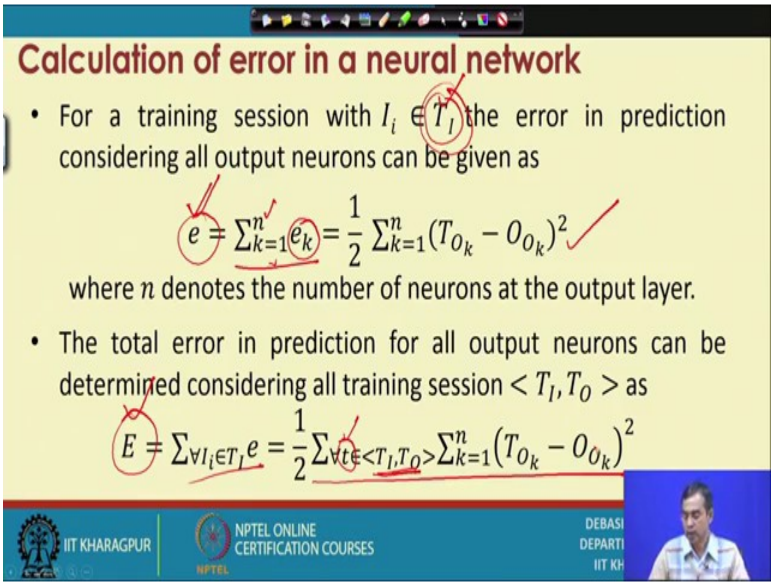


Now, so, we are basically discussing the calculation of ΔV and ΔW, that means, the change or updated values of V and W matrix. Now, as I told you it basically based on the error calculation now, let us see how the error can be represented. Now, we will consider any neuron let it be k-th neuron at the output layer. Now, for any input in the training set we decide that say I suffix i denotes the input which is there in training set it is a i-th input that belongs to the input training set and as you have already mentioned that this training set T is decided by T_O and T_I. So, $I_i \in T_i$ and corresponding to this I_i there is one O_i which is there in T_O.

So, it is basically the observed output due to the i-th input which is there in that T_O sets. So, this is basically the target output there. Now, we can this way we can write say T_{O_k} denotes the target output; that means, the true output of the k-th neuron which is there in T_O sets. So, this is the notation that we will follow it and then once we know the true output and then observed output of the k-th neuron in the output layer we represent or the back propagation algorithm represents the error of the k-th neuron is by this formula, it is basically average of the difference the square of the difference of the true output and then observed output.

So, true output and then observed output. So, e_k is basically is the formula to calculate the error of the k-th neuron. Now, so, this way for all input that is there in T_I can be known and hence the T the true output also can be note and therefore, for each input and then for each perceptron the error can be known the error can be calculated. Now, this error can be used to calculate the error function e.

(Refer Slide Time: 18:34)



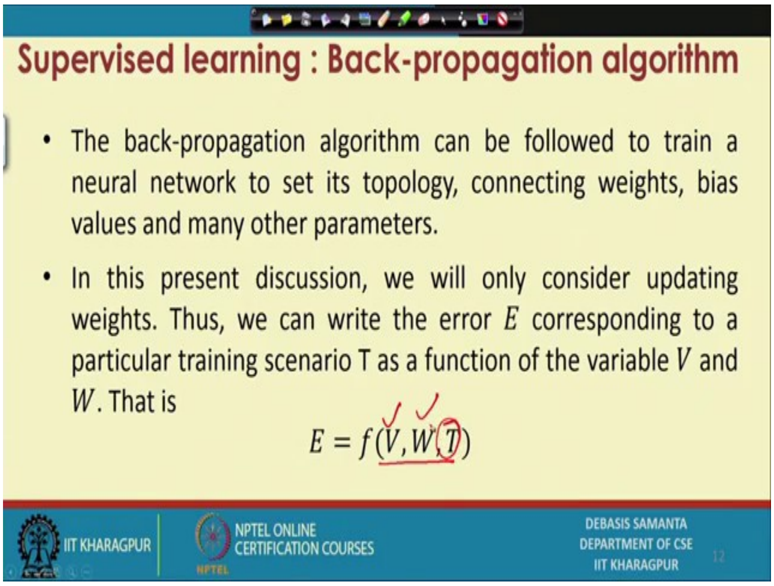
Now, we will see the error function e here. So, here basically the total error e it is denoted as this e, that means, it is summing up for the errors from the all neurons that is there in the output layer. So, $\sum_{k=1}^{n} e_k$ basically the error all errors this basically calculates all errors that can be obtained for a given training for a given input that belongs to a training sets and then in sum.

So, this is basically the summation form nothing, but. Now, this is the error for one input which is there in T_I. Now, T_I contents there are many other inputs also. So, considering all the inputs which is there in T_I and if we take the sum of all the errors that can be obtained, then it will give you the error. So, this basically, the formula of error function that can be obtained which can be written in this form.

Where, T is basically the training sets that is there, for all T belongs to for all inputs that is belongs to these sets. T is basically a training data that can be belongs to these sets and taking the summation of all the neurons for all inputs and then this one then the error function can be calculated. So, what I want to say is that given a training set T_O, T_I and knowing the knowing the output of the network we shall be able to calculate the error function at any instant and as I told you the Steepest descent method is basically is to find some values of the neural network parameters, for example, V and W parameter in our consider case so that this error value will be minimum.

So, now, we have learnt about how the error of a network can be calculated. Now, this error calculation is a one important thing and now, we will see how this error calculation is useful and then the back propagation algorithm it is.

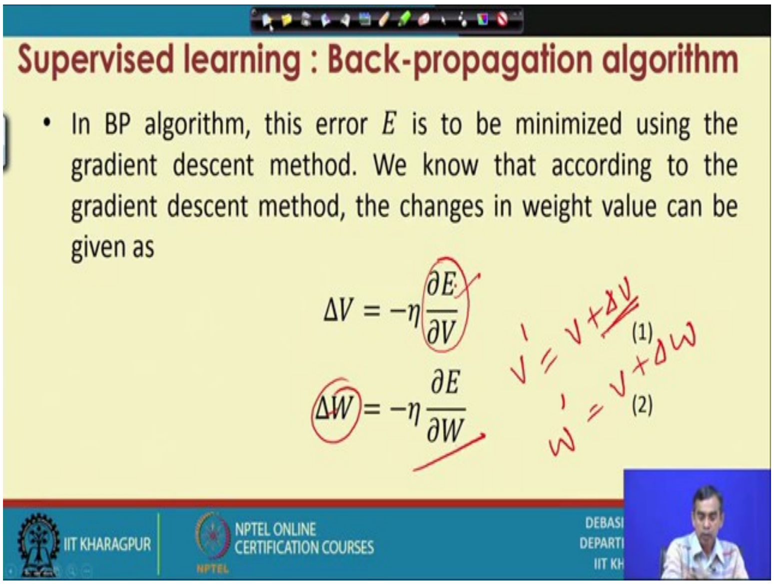
(Refer Slide Time: 20:43)



Now, so, in summary what we can say is that in summary the error E is basically a function of error E is a $f(V, W)$ and then training sets and then how this function looks like we have discussed in the last slides. So, for given values of V and W and then for the given training sets, E can be calculated. So, in our again I repeat in our procedure or in the back propagation algorithm is to find the V and W value in such a way that for this training set T the error E is minimum.

So, it is basically minimizing error and finding the V, W and hence the neural network will learn it.

(Refer Slide Time: 21:53)

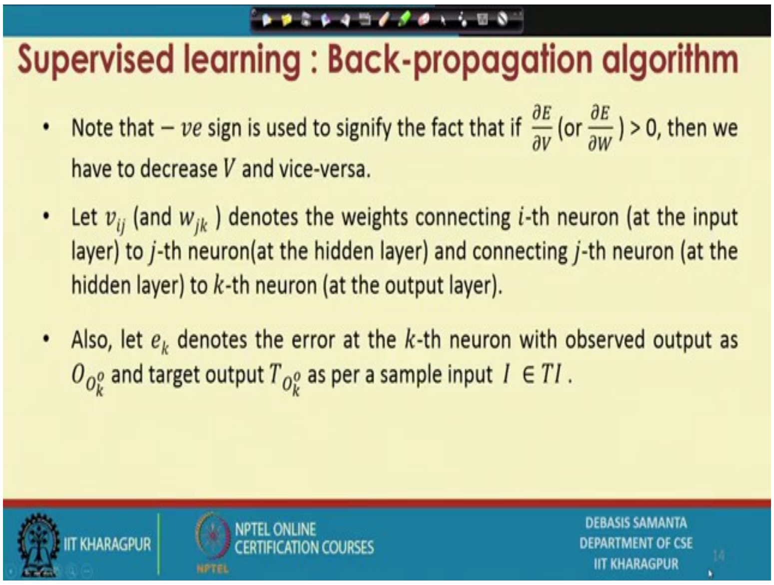


So, essentially it is a optimization problem, that means, how the V and W value can be considered. Now, let's come to the discussion of how this concept that mean error can be back propagated and then back propagation algorithm. Now, here is the idea about we have discussed about delta rule. So, this is basically the delta rule that can be followed there. Now, we can note that one negative number is used. Now, this negative number is for the function is that if error is increases then this value needs to be decreases.

So, that is why if it is increases and it decreases so, the opposite sign is used. On the other hand, if it is decreases then it values to be increases so, negative sign is there. So, that is a convention that it will be there. Now, once this ΔV value is known to us then the next value V' is basically (V+ΔV) value. So, this ΔV value will be either incremented or decremented that depending on this what is called the slopes or gradient or first order derivatives if it is increases then it should be decreases if it is decreases it should be increase and vice versa.

Similarly, W' the next the weight value if at any instant the V value is there then ΔW where ΔW can be calculated using this formula. So, these are the Steepest descent method by which how E changes with V and then knowing this changes how the updated value updated value can be considered and then the revised value of the V or W matrix will be obtained. So, this is the idea that is followed here in a Steepest descent method.

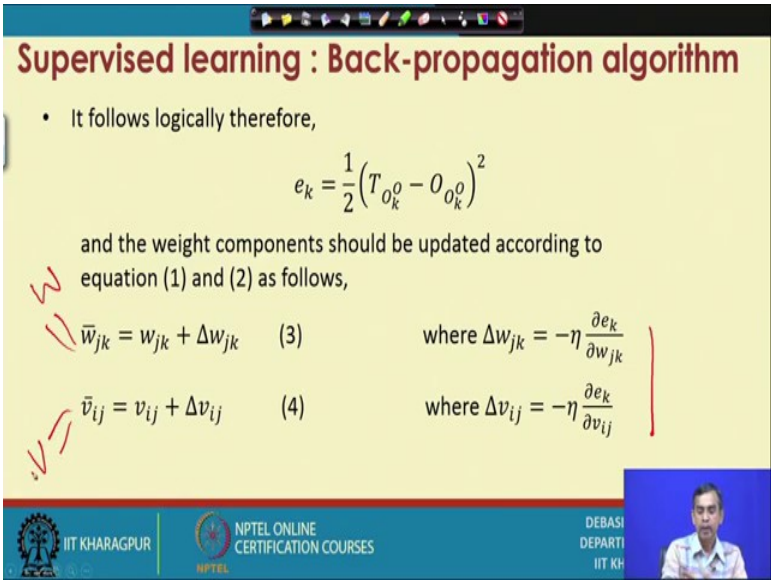
(Refer Slide Time: 23:42)



And, the delta rule is the most important useful is there as I told you the negative sign is to signify the fact that if $\frac{\partial E}{\partial V}$ the first order derivatives of error with respect to V or $\frac{\partial E}{\partial V}$ > 0, then we should decrease V and vice versa. And, v_{ij} this is our usual notation that we have discussed about v_{ij} and w_{ij} denotes the weight values connecting from the i-th neuron in the input layer to the j-th neuron in the hidden layer and then from the j-th neuron in the hidden layer to the k-th neuron into the output layer.

And, e_k denotes the error at the k-th neuron which is observed output is a difference sum is basically half of the sum of the different square of the observed output and the true output for each input i.

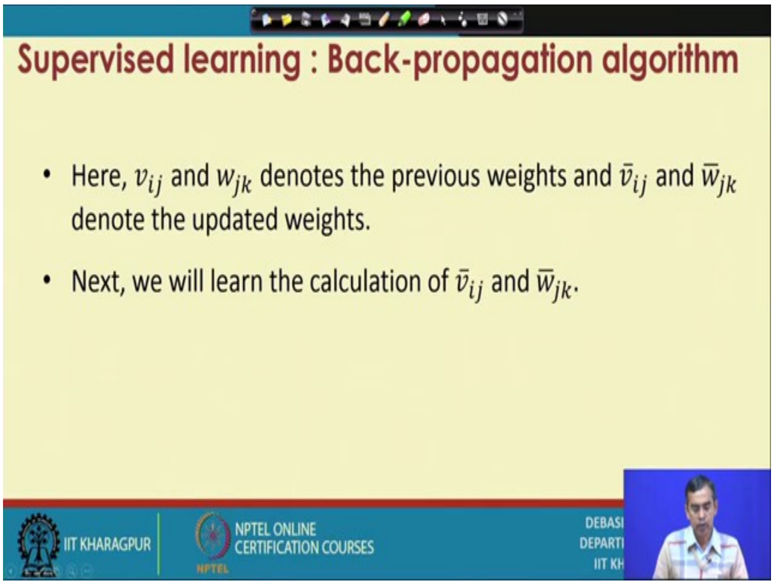
(Refer Slide Time: 24:54)



Now, now so, this is the concept that is there and ok, with this concept we will be able to learn about back propagation algorithm quickly. Now, as I have all learned about so, e_k is basically the error of the k-th neuron in the output layer and we know that it is the formulation and if we know the error of the k-th neuron in the output layer then we shall be able to know what is the in incremented value of the Δw_{jk} . So, it basically if this is the current value of w_{jk} or current value of v_{ij} , then using this delta rule we will be able to calculate Δw_{jk} and then updated value Δ is a updated value this one. Similarly, using this rule we will be able to calculate this one and then updated value will be there.

So, this is the delta rule and using this delta rule we will be able to calculate the updated value and if we know any j-th and k-th weight or i-th and j-th weight we will be able to know them for the entire V and W matrix because this is basically give the W matrix and this is basically gives the V matrix. So, this is the idea that is followed in the back propagation algorithm and once the error is known to here then we will be able to find the updated value.

(Refer Slide Time: 26:10)



Now, how the updated value can be calculated. Back-propagation algorithm suggests one very clever method, this method is called the chain formula or it is called the chain formula in a back way. So, is starting from the input a starting from the output then go to the next previous layer in the next layer in the previous layer and so on so on. So, we will discuss about this is the propagation algorithm that is why it is called a propagation algorithm and it is propagation from the back.

We will discuss about this back proportion algorithm in the, next session, ok. So, back propagation algorithm we will be discussing the next session and we learn about how that network can be learned tool to if basic network can be trained to learn the values of V and W parameters values.

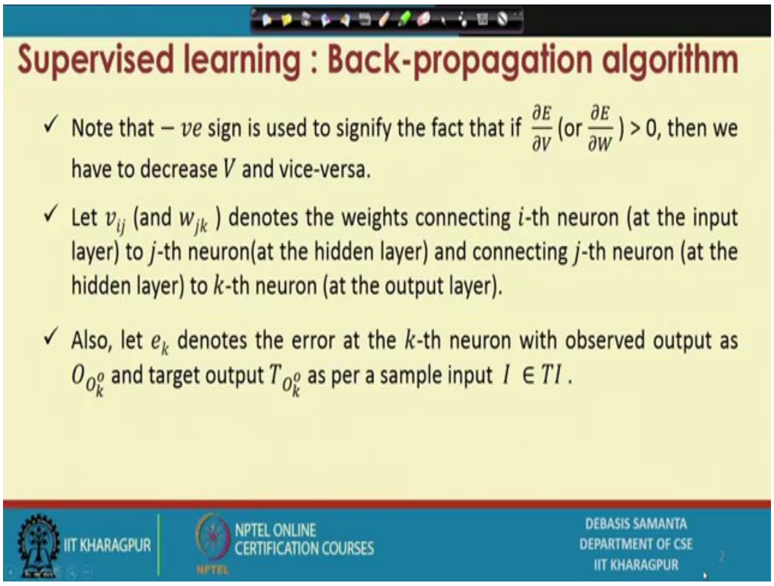
Thank you.

Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 39
Training ANNs (Contd.)

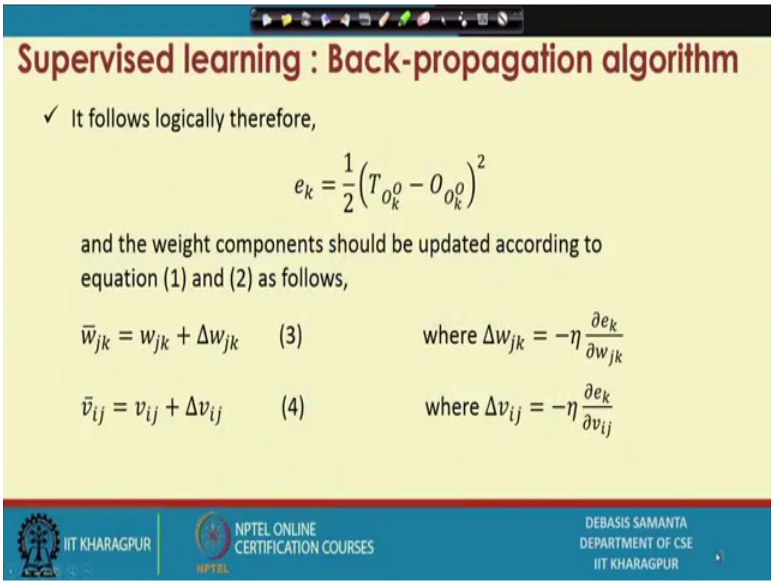
So, we are discussing about training, multilayer feed forward neural net and training procedure that we are going to follow is basically supervised learning and in a special case of supervised learning we are learning back-propagation algorithm.

(Refer Slide Time: 00:39)



Now, in the last lectures, we have discussed about the chain rule based on the method of gradient descent. So, chain rule basically tell about if we can compute the error of any neuron and then using this error calculation how we can decide the updated value of the neural network parameters. So, these are the concept that we have learned in the last slides.

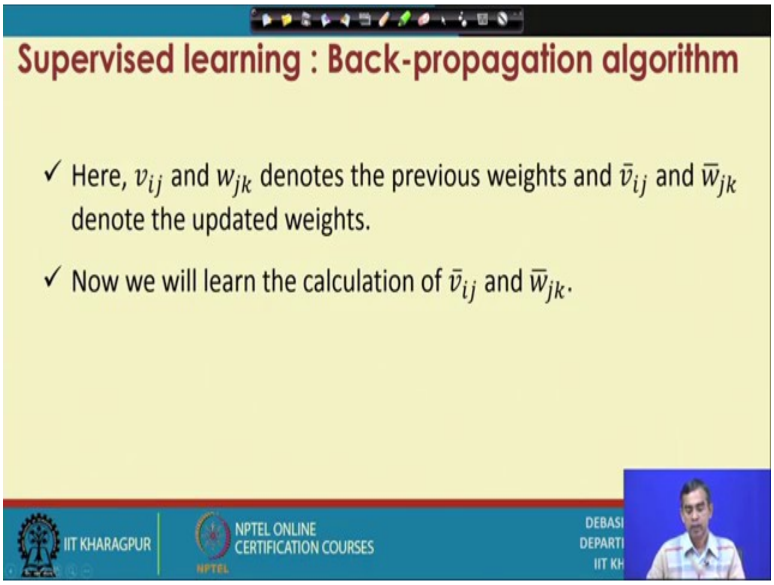
(Refer Slide Time: 01:01)



And, also we have discussed about how the error can be computed and thereby what will be the delta rules. So, the delta rule that we have discuss in the last is these are the delta rule. So, the delta rule is there and based on this delta rule the updated values of the neural network parameter that can be obtained.

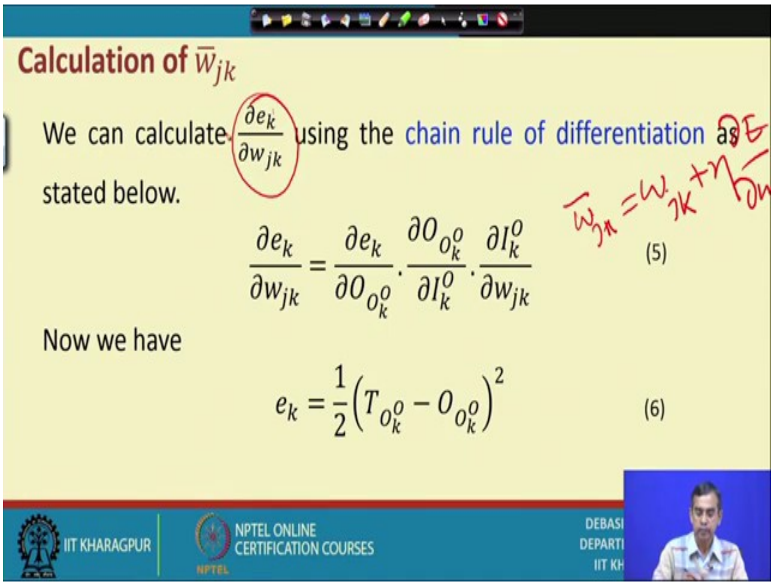
Now, that things we have these are the things we have discussed in the last lectures.

(Refer Slide Time: 01:28)



Now, in this lecture we will discuss about how the updated value can be computed and there is a systematic method or step and this systematic step is basically called the back-propagation algorithm which is a main point of discussion in today's in this lecture.

(Refer Slide Time: 01:52)



Now, so, back-propagation algorithm, it is basically the idea about is that. Ok, first we will discuss about the calculation of the chains weight values in between the j-th neuron in the hidden layer to the k-th neuron in the output layer. So, it basically we have to calculate w_{jk} , this is updated value, this is equals to w_{jk} , the previous value, plus it is basically and $\frac{\partial e_k}{\partial w_{jk}}$. So, this is the things that we have to consider it and this basically to essentially this is a calculation of $\frac{\partial e_k}{\partial w_{jk}}$. Now, let us see how this calculation can be obtained.

Now, here so, basically how the error is increases with the increased of the values of w_{jk} , if it is represented by $\frac{\partial e_k}{\partial w_{jk}}$, then we can say that this error e_k is basically represented by the output of the output of the k-th neuron in the output layer. So, it is basically e_k is a function of output of the k-th neuron in the output layer. So, it is basically $\frac{\partial e_k}{\partial k}$.

Now, this output of the k-th neuron in the output layer again is depends on the output of the k-th neuron in the input layer. So, it is basically we can denote that I_k^o . I_k^o denotes that what is the input to the k-th neuron at the output layer. So, input to the k-th neuron at the output layer is basically influenced the output of the k-th neuron at the output layer and now, again this values; that means, input to the output layer k-th neuron of the output layer is basically see influenced by the weight of the weight from the j-th layer to the k-th j-th perceptron in the hidden layer to the k-th layer.

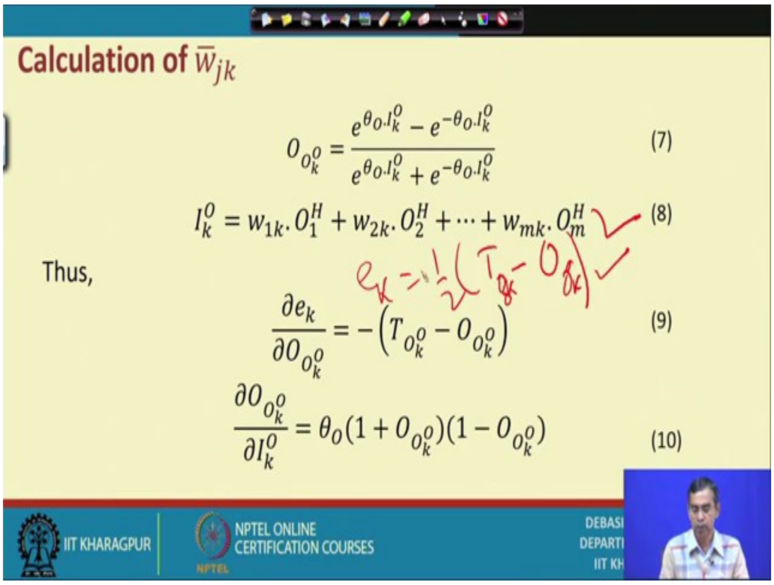
Now, so, this basically is the idea about this is the chain rule. So, if we know this then this and then this. So, it is basically chain rule. Now, this chain rule chain rule of differentiation rather, is basically to compute the error and then the relation between error and w_{jk} . I hope you have understood this chain rule of differentiation, it basically gives is basically the chain chaining of differentiation or a dependency parameters; what I can say in other words if we move from input to the output direction.

So, the w_{jk} will influence the input, this input will influence the output, this output will influence the error and therefore, it is basically $\frac{\partial e_k}{\partial w_{jk}}$, that means, how the error is influenced by the this one. So, error is basically influenced by this one, but it is in the form of a chain. So, this chain can be propagated in a back direction, it is basically output to the input to the w value. So, it is called back-propagation.

Now, so, this form this calculation is easy once we know the e_k value. e_k can be obtained in terms of true output and then the observe output. So, this differentiation also can be calculated if we know these functions.

Now, we will see exactly how this can be calculated in a more mathematical way or in a very systematic way. So, so this is the chain rule of differentiation and I have discussed the chain rule of differentiation for the w_{jk} calculation or updated value of w_{jk} calculation.

(Refer Slide Time: 05:59)



Now, we will discuss about more details that how this w_{jk} value can be calculated as we know the output for the perceptron in output layer we also know what is the input to any k-th perceptron in the output layer and also we know what is the output from any perceptron in the hidden layer and then this one. So, basically here all the values if it is known to us then we will be able to calculate this differentiation form first one is derivatives e_k with respect to w_{jk} .

Now, let us follow. So, this basically we have already learned about. So far the output layer computation that we have discussed there this basically denotes the output of the k-th neuron at the output layer. So, it is basically output of the k-th neuron at the output layer and it is the function that it is here where I_k^o denotes the input to the k-th neuron at the output layer.

Now, and θ_O is basically the threshold function of the perceptron in the output layer. So, this is the formula for deciding or a representing output of any k-th perceptron in the output layer and then the I_k^o this is the output layer computation where the input to the output layer we have computed and in this output layer computation.

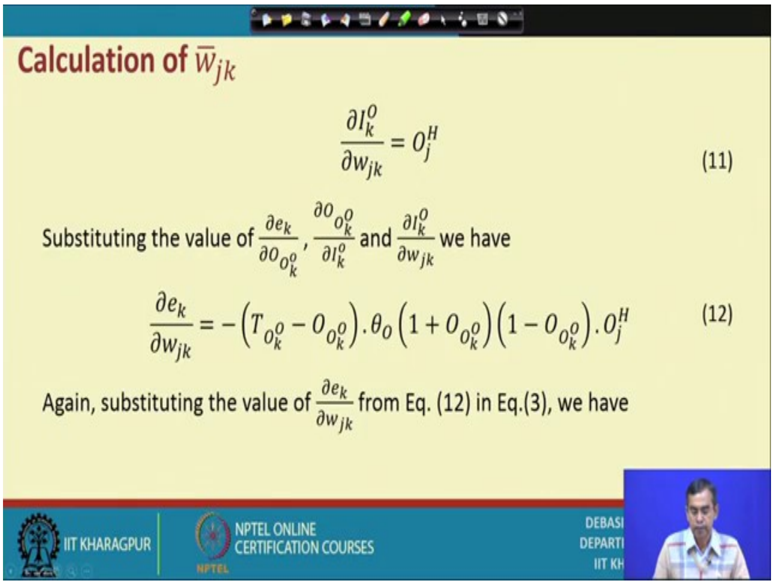
We know that this is the input to the k-th neuron at the output layer and which can be expressed in terms of this summation sum of the product form where w_{lk} is basically the weight value from the neuron perceptron one to the k-th perceptron the this is a perceptron 1 in the hidden layer to the k-th perceptron, perceptron 2 in the hidden layer to the k-th perceptron and corresponding to the output of the hidden layer. So, this is basically output of the first perceptron in the hidden layer, output of second perceptron in the hidden layer and so on so on.

So, this I_k^o therefore, can be obtained by this one and you have discussed about how the matrix representation of the same thing is there we will see exactly how this matrix representation can be ultimately to be used. Then using this formula so, first we have this $\frac{\partial e_k}{\partial O_k}$ and e_k is basically e_k that we have discussed about $\frac{1}{2}\left(T_{O_k^o}-O_{O_k^o}\right)^2$. Now, this $\frac{\partial e_k}{\partial O_k}$, that means, it taking this first order derivatives of this one we will be able to obtain this formula. So, this is basically $\frac{\partial e_k}{\partial O_k}$ calculation.

Now, so, for our chain rule of differentiation, this is the first calculation and then the second differentiation is $\frac{\partial O_k}{\partial I_k}$. So, this can be obtained by differentiating this expression with respect to I_k^O. Here, I_k^O is a parameter, if we differentiate this one O_k^O with respect to I_k^O then we can see after a lot of simplification this expression can be obtained. So, $\theta_O\left(1+O_{O_k^o}\right)\left(1-O_{O_k^o}\right)$.

So, the detailed calculation I have avoided you can try yourself. So, that from these things we can represent this expression. So, this basically gives the, this is the first differentiation in the chain rule of differentiation. It is the second differentiation in the chain rule of differentiation for w_{jk} and then finally, we will be able to obtain $\frac{\partial I_k^O}{\partial w_{jk}}$.

(Refer Slide Time: 09:41)

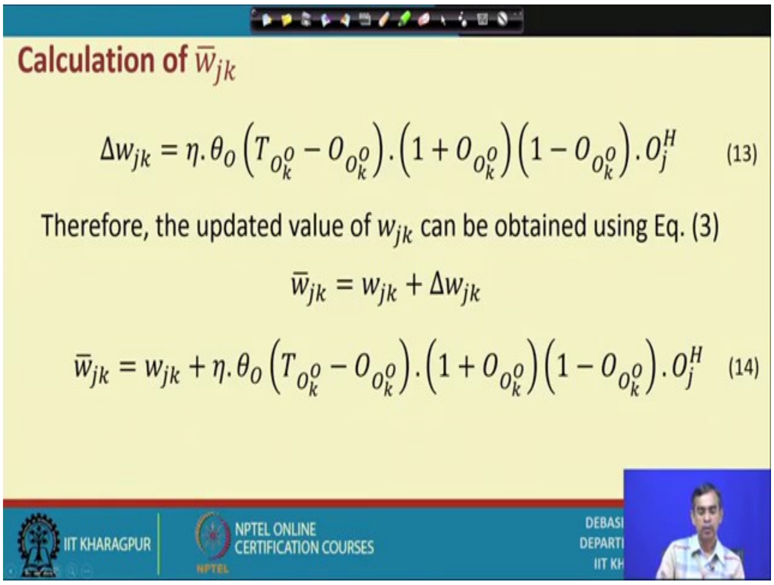


So, it is the formula it is there. Here $\frac{\partial I_k^O}{\partial w_{jk}}$ is basically O_j^H. So, O_j^H, that can be obtained from the previous calculation, ok.

So, now, we have learned about that the differentiation values of these, this and this one and then combining this by means of chain rule or differentiation we can obtain this one the expression. So, this basically gives the calculation of $\frac{\partial e_k}{\partial w_{jk}}$, that means, what will be the error at the k-th perceptron if we change the weight value between the j-th and k-th perceptron in the two layers, hidden layer and perceptron layer. So, it is basically the expression that can give you the value of this change.

Now, this value when you substitute, it to the modified value of this one we will be able to calculate the modified value.

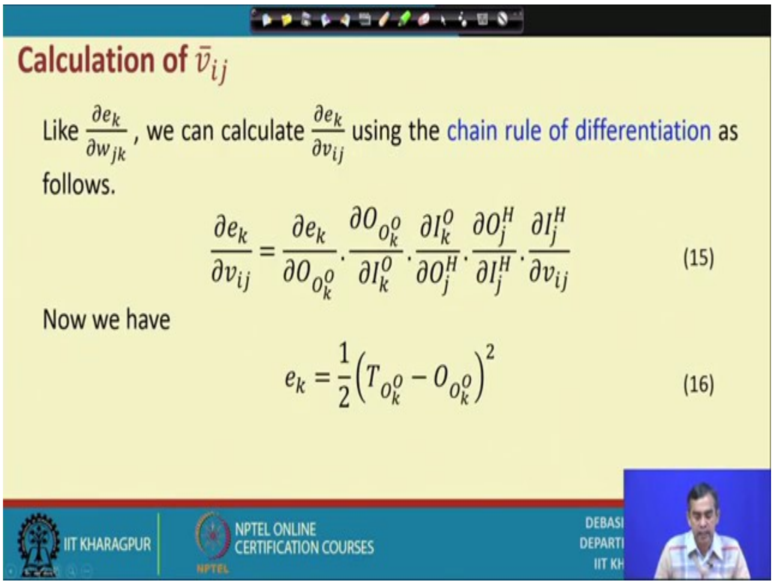
(Refer Slide Time: 10:43)



So, modified value will be like this. So, Δw_{jk} it is basically the modified value, Δw_{jk} it is the modified value and this basically using the chain rule of that is the delta rule. So, $\eta.\frac{\partial e_k}{\partial w_{jk}}$. So, once we know this incremented value we will be able to obtain the updated value and this is the updated value.

So, straight away we can write this is the updated value is basically if this is the current value and this is the chain value and then updated value can be obtained. So, you can see that all the updated value can be obtained in terms of training data and this training at the this is the true output and this is the observe output, observe output is basically decided by the v and w value as well as the input.

(Refer Slide Time: 11:45)



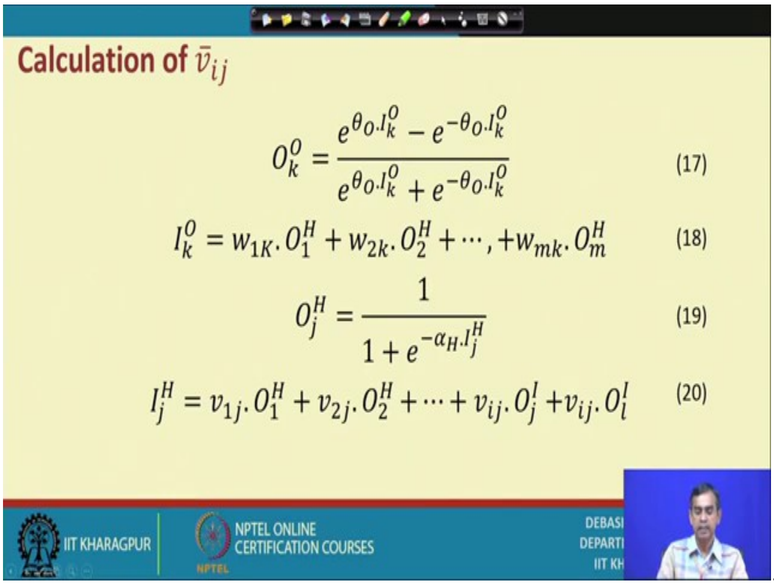
Now, we have learned about how the updated value so far the w matrix is concerned now, we will discuss about the same calculation, but for the v matrix. So, v matrix calculation is like this, it is the same way the chain chaining rule of differentiation that we have followed to calculate the w_{jk} we will follow the same rule, but little bit different way. Now, let us see first discuss about the chain rule of differentiation with respect to v_{ij} ; that means, here e_k error k is influence when the v_{ij} changes; that means, $\frac{\partial e_k}{\partial v_{ij}}$ gives basically if we change the v_{ij} then how the error will change and this change can be expressed by means of chain rule of differentiation.

Now, here again see the chain rule of differentiation. It is basically, we want to find how the error will change if we change the v_{ij} value; that means, the weight values from the i-th perceptron in the input layer to the j-th perceptron in the hidden layer. Now, here this e_k as we know is basically depends on output. Now this output again depends on input to the output layer. This input to the output layer depends on output of the hidden layer. Now, output of the hidden layer depends on input of the hidden layer and this input of the hidden layer depends on the values of the v_{ij} .

So, it is basically ∂e_k . So, ∂e_k is a function of ∂v_{ij} by it is basically chain from output layer of the perceptron in the output layer. Output of the output layer to the input to the output layer to the output of the hidden layer to the input to the hidden layer and finally, to this v_{ij} . So, this chain rule is basically here and the differentiation takes place this form.

Now, we have the all values for example, we have already known the values of O_k^O , I_k^O then O_j^H and then I_j^H . So, knowing all these values we will be able to calculate all the differential form. Now, let us see how the differentiation form can be calculated. So, far the $\partial e_k, \partial o_k$ is concerned we will use this form this is the error calculation and differentiating this e_k with respect to O_k , we will obtain this one. Now, for this one we will consider I_k^O . Now, let us see how the I_k^O can be represented that we have already representing when we are discussing about the hidden layer computation.

(Refer Slide Time: 14:14)



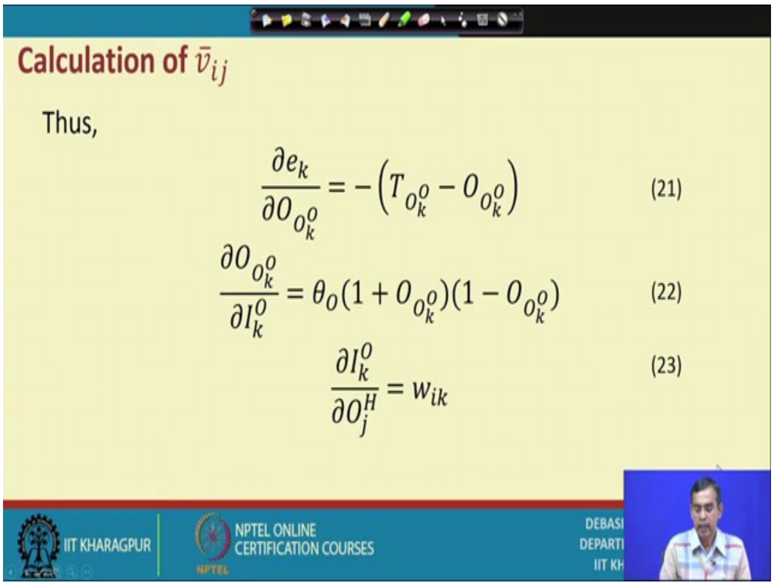
So, hidden layer computation says that this is the I_k^O or this is the basically input layer combination of the output, right. So, input combination computation in the output layer computation that we have discussed. So, it is I_k^O . I_k^O it can be expressed this one.

Once knowing this I_k^O and then we will be able to calculate that is $\frac{\partial O_{O_k^o} I_k}{\partial I_k^o}$.

Now, so, so this basically expression is useful for the second differentiation in the chain rule for the v_{ij} . This expression is required to know the third differentiation in the chain rule and this is the expression that is required for the first differentiation in the chain rule and finally, this is the expression that is required for the last difference in the chain rule.

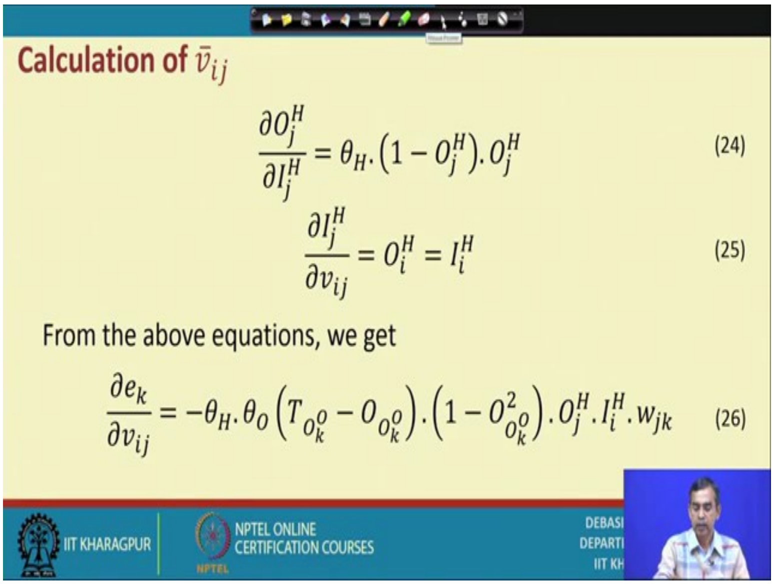
So, all the expression that can be obtained can be used to calculate the final value of $\frac{\partial e_k}{\partial v_{ij}}$.

(Refer Slide Time: 15:14)



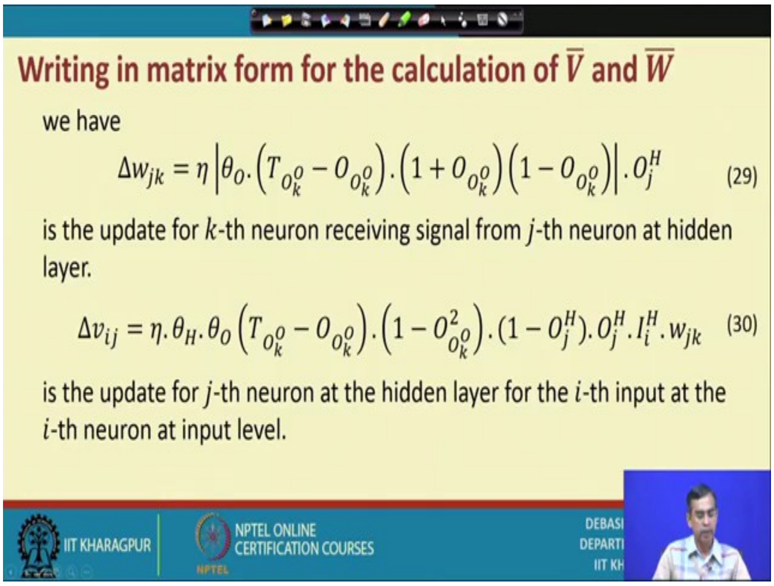
Now, here is the total composition it is like this. So, $\frac{\partial e_k}{\partial O_k^o}$ can be obtained from the first rule. First what is called the e_k versus the output this one and so this one and this $\frac{\partial O_k}{\partial I_k}$ can be obtained from the input of the output layer computation and this is basically the output of the hidden layer combination and.

(Refer Slide Time: 15:49)



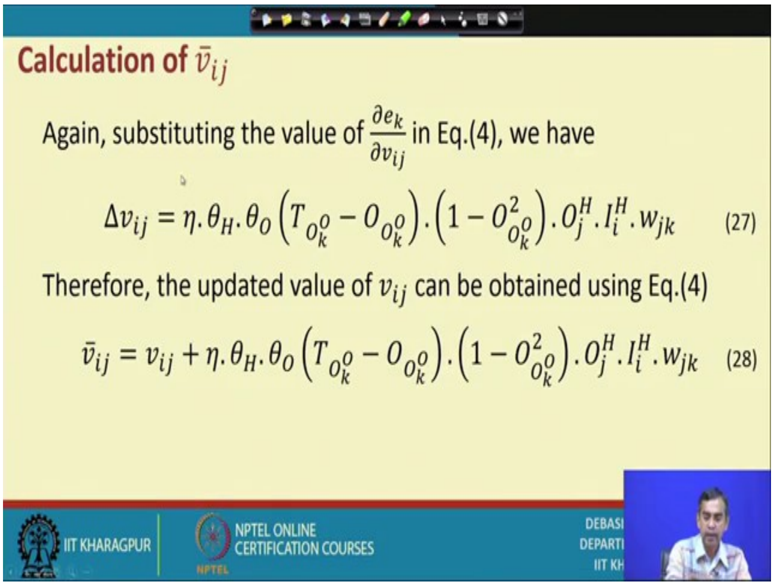
And, this is basically the computation from the this is basically the computation this is basically computation from the hidden layers and finally, this is the computation with respect to v_{ij}. Now, all this expression can be obtained and then finally, putting all the calculations in the chain rule of differentiation ultimately we will be able to calculate this $\frac{\partial e_k}{\partial v_{ij}}$ and which takes this form like this. So, this is the final form represented with the calculations of all the differentiations for the different layers computation.

(Refer Slide Time: 16:34)



So, this basically the way and then once we know this values $\partial e, \partial v$, $\frac{\partial e_k}{\partial v_{ij}}$ we will be able to calculate the updated value, updated value v_{ij} .

(Refer Slide Time: 16:37)



So, finally, using the chain using the delta rule of steepest descent method we will be able to calculate this is the Δv_{ij} . Once the Δv_{ij} known then the modified values of the weights from the i-th to j-th neuron can be expressed this one.

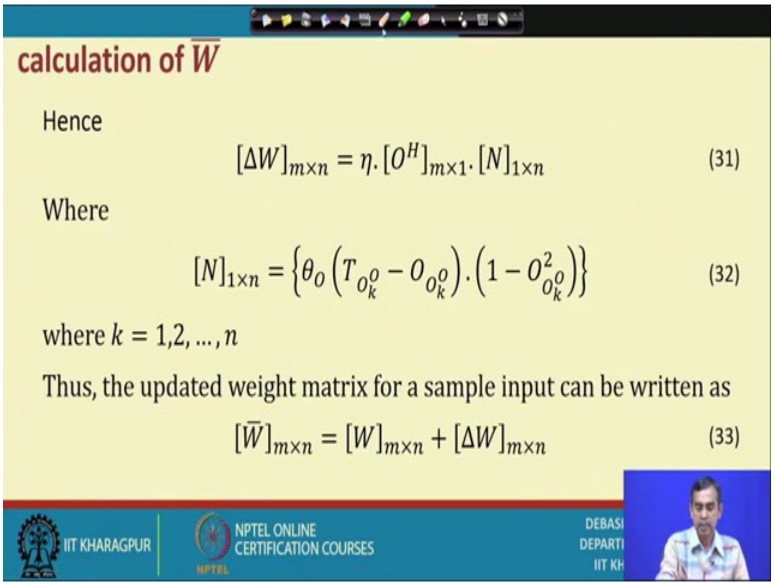
So, we have learned about with respect to some training input. Then how the v_{ij} and the w_{ij} can takes it change values. Now, this change values is basically following the steepest descent method that mean it will decide the next value so that it will minimize the error.

So, this is the back-propagation algorithm which follow the steepest descent method are there now. So, these methods that we have discussed about calculation of the differentiation values at every neuron in the network, but at a every neuron network if we do one by one and if there are large number of neurons in the network then it is computationally infeasible.

So, in order to make this I mean sorry address this problem so, there is a method by which the entire thing can be expressed in the matrix representation. Now, here so far the matrix representation is concerned. So these are the calculation can be represent in one matrix these are the calculation can be represented one matrix and then finally, whole W can be represented one matrix. So, this is for the w_{jk} similarly, for the v_{ij} .

So, this is the one matrix representation this is the another matrix representation and putting all the things together a compact matrix representation can be obtained. I will not discuss about details computation, but at the final result I will explain that how this matrix form can be obtained for the entire W and V matrix here.

(Refer Slide Time: 18:44)



So, this is basically a little bit careful calculation a lot of steps if we can follow, you will be able to derive into this one where O^H in our previous discussion we have discussed that it is basically the output of the hidden layer. So, it is in the form of matrix $m\times n$ and n is the one matrix of the size $1\times n$, where n matrix is there in the previous expression that is there. So, $[\Delta W]_{m\times n}$ that means, it will take care for any w_{jk} from any j-th neuron in the hidden layer to the k-th neuron in the output layer.

So, this basically gives the matrix representation of the error changes or updated values of w_j , that means, this will take all the updated values all the set, but for this we need only the output layer and then is multiplied by this form for every neuron. So, this can be obtained for $k=1$¿n it is basically one row column matrix like.

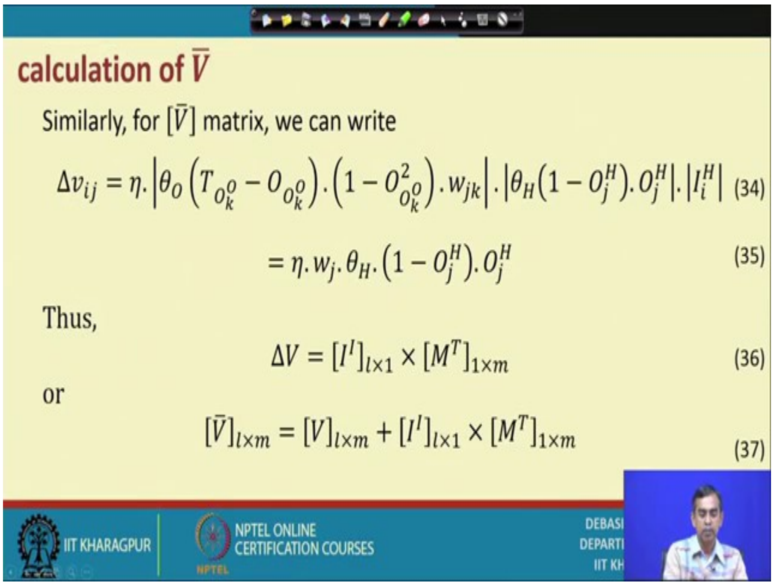
So, this is one column matrix and this is one row matrix and it is a product of row matrix and column matrix for all neurons that is there in the output layer in between hidden and output layer and it will calculate the $[W]$ matrix in between hidden layer and output layer. So, this is basically is a compact matrix representation in terms of two matrix $[O^H]$ and then N ¿], where $[N]$ is this one that $[\Delta W]$ matrix can be obtained.

So, this is the idea about $[\Delta W]$ matrix, once the $[\Delta W]$ matrix is known we can know if this is the $[W]$ matrix at any instant and this is the updated matrix according to this formulation then the updated matrix $[\acute{W}]_{m \times n}$ can be obtained using this one. So, if it is known to us knowing this one we will be able to calculate what is the next value.

So, this is the concept and here I can say again that simple matrix calculation wants the different values of the neurons in different layers are known we will be able to find it quickly and then this matrix can be obtained. So, this way the network can be learned for the next matrix.

Now, this is the case of the $[W]$ matrix. The similarly, the $[V]$ matrix also can be expressed.

(Refer Slide Time: 21:12)

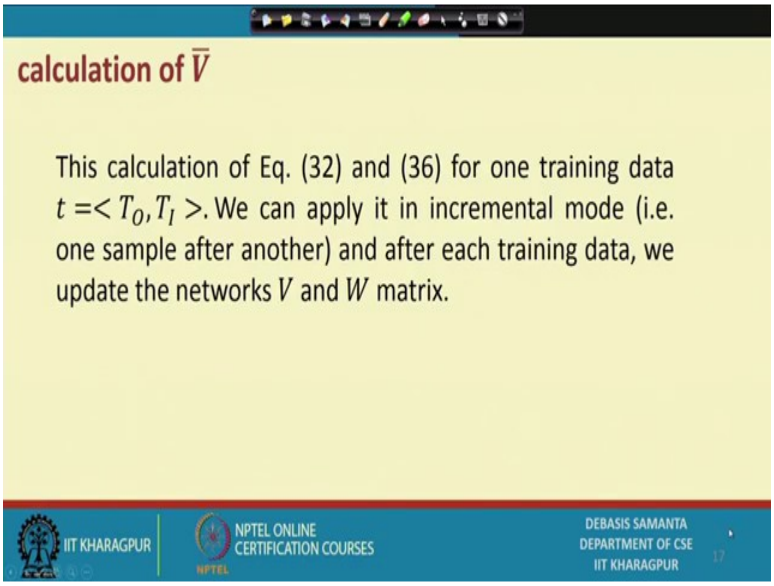


In case of $[V]$ matrix as you have already discussed about. This is the v_{ij} calculation Δv_{ij} which can be represented in this form. So, it is basically this one and this one like. So, this basically input $[I^I]_{l \times 1}$ and this is basically M^T , M^T , is this one. So, Δv_{ij} can be expressed this way and finally, this is the updated matrix

V ¿] just like a $\acute{W}$ and this is the current one and this is the implementation one and we will be able to calculate it.

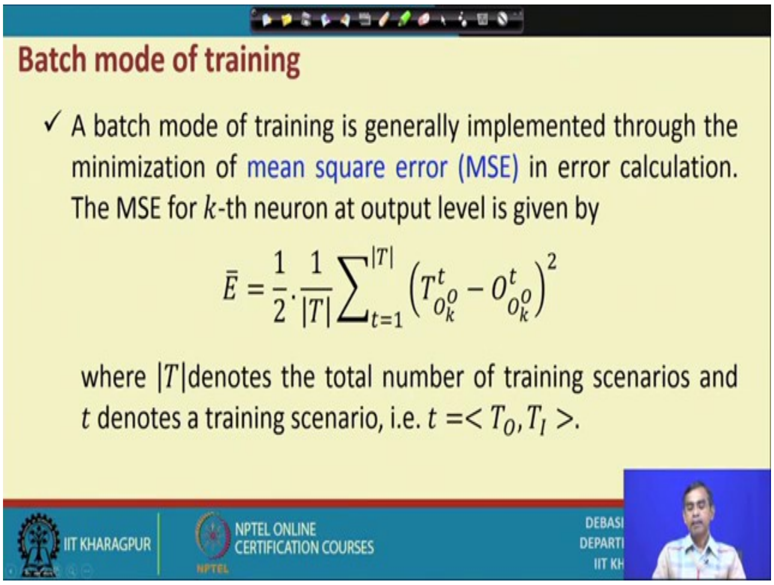
So, there are some steps has been jumped here so that we can give finally, so finally, this is very important one expression that if we know this matrix at the moment and if we know the input to the system and $[M^T]$ matrix which is basically this is the $[M^T]$ matrix, $[M^T]$ matrix for all neurons in the k-th layer. Then we will be able to calculate the $[\acute{V}]$ matrix. So, it is just a matrix calculation so far the calculation of updated matrix is concerned. So, both W ¿] and then V ¿] matrix therefore, can be calculated using simple matrix product operation.

(Refer Slide Time: 22:30)



Now, so, this way for the different input the output if we know and then we will be able to calculate the V ¿] and then W ¿] matrix there.

(Refer Slide Time: 22:35)



Now, here is idea about it that we have considered about training data. In a training data there may be a large number of inputs and then outputs. So, all the training data can be processed in a different way, because if the training data is very large. Then we can apply this training data in the batch mode. So, we can basically consider out of the entire training data a sub set of training data. Then this subset can be applied and then it can be tested and then the V and W can be decided. Then some other subset can be used for the test data. So, we apply it and then calculate this one and then again we will see that how much error is there and if we repeat the same procedure again and again then we will be able to calculate the errors and then finally, the V and W the final form.

So, this is called the batch mode of training. There are many training the strategy is known to us like say cross validation method or is a k fold validation method like this, but those things are not possibly discuss in this in this discussion in this lecture.

Anyway so, I have just given an idea about how a network can be trained. So, that it can learn the network parameter V and W in this case the same idea which basically we have followed to learn the V , W can be extended to learn the other parametric values like say number of neurons in the hidden layer number of hidden layers in the hidden layers, number of layers in the hidden level then the transfer function.

We have considered that these are the transfer function that we have to follow in each layer, but that can be also can be updated the different transfer functions for different layer and then again see for which transfer function values it gives the better result that can be decided finally, after a lot of trial and error method.

So, the procedure it is there, but the programming environment which is required to find the good network to solve your problem for a given training data is seems to be very tedious job, but with the help of tool it is very simple job on. So, in the next lectures, we will discuss about the different tools which are available which can be used to solve many problem using the different soft computing approaches like fuzzy logic, the genetic algorithm and neural network.

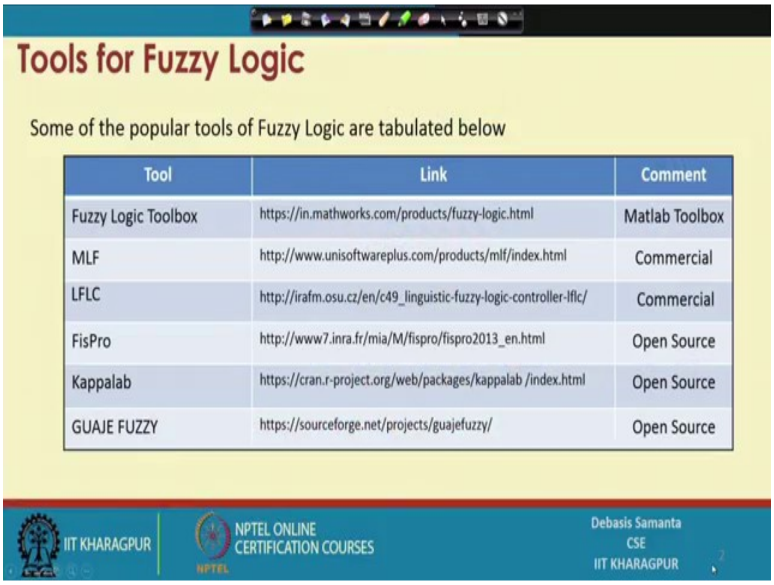
So, in the next in the next lecture we will discuss quickly about the different tools that is available in our so far the different tools that is available and how to use the tools all the different problems, ok. Thank you very much.

Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 40
Soft computing tools

Yes, we are almost at the end of this online NPTEL course on introduction to soft computing. This is the final lectures. In this final lecture, we will learn about the different tools which are available to solve, the problem using the different soft computing techniques that we have learn in the course. In this course, we have covered mainly three soft computing paradigms. One is fuzzy logic, another is genetic algorithm, and the neural network. Now so, plan of this lecture is basically to cover the different tools which are available in the market, and then its application how those tool can be used.

(Refer Slide Time: 01:06)



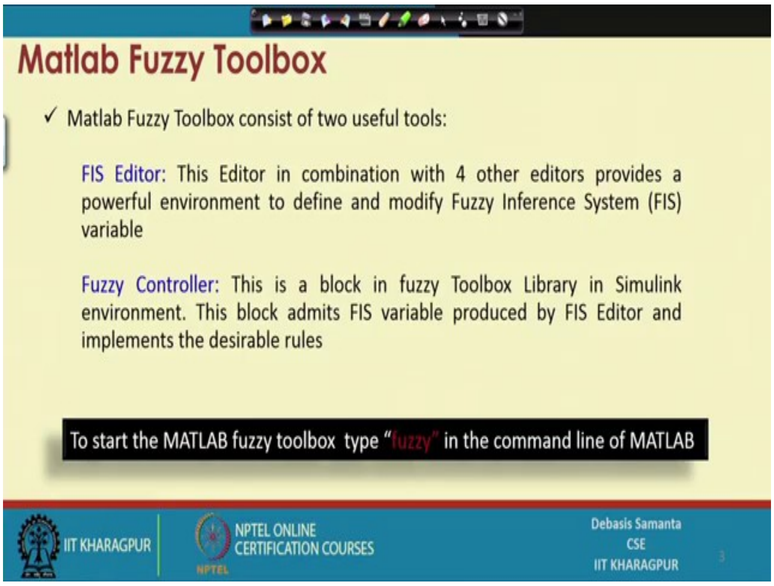
Tool	Link	Comment
Fuzzy Logic Toolbox	https://in.mathworks.com/products/fuzzy-logic.html	Matlab Toolbox
MLF	http://www.unisoftwareplus.com/products/mlf/index.html	Commercial
LFLC	http://irafm.osu.cz/en/c49_linguistic-fuzzy-logic-controller-lflc/	Commercial
FisPro	http://www7.inra.fr/mia/M/fispro/fispro2013_en.html	Open Source
Kappalab	https://cran.r-project.org/web/packages/kappalab /index.html	Open Source
GUAJE FUZZY	https://sourceforge.net/projects/guajefuzzy/	Open Source

I will give an idea about it and finally, I will discuss about some hybridization approach. Now, let us first discuss about the different tools which are available to solve some problem using fuzzy logic. Now, there are many tools of course, the some tools are available as a open source. So, you can just simply download the source code, and then use it as an open source. The tools which are available in the market, we have discussed here.

So, these are the last three tools are open source is called the FisPro, then Kappalab and another is that GUAJE FUZZY. So, GUAJE FUZZY is developed for Japanese scientist, Kappalab is also from Chinese scientist, and FisPro is a open source is a fuzzy group. So, these are the different open source tool that is available there. Other than the open source, there are many sophisticated tools are available in the market to use the fuzzy logic concepts. So, these tools are called the MLF and LFLC. Now, these are the commercial tools a lot of features are there, and it is very useful. Now, other than this another commercial tool, another one commercial tool is also available which is very popular it is called the Fuzzy Logic Toolbox.

The fuzzy logic toolbox is available in Matlab software. So, this is the fuzzy logic toolbox is popular, and I will discuss about the fuzzy logic toolbox, how this can be used to solve some problem using fuzzy logic. So, so, this is the fuzzy logic toolbox that is available in the Matlab toolbox we will discuss about it.

(Refer Slide Time: 03:01)



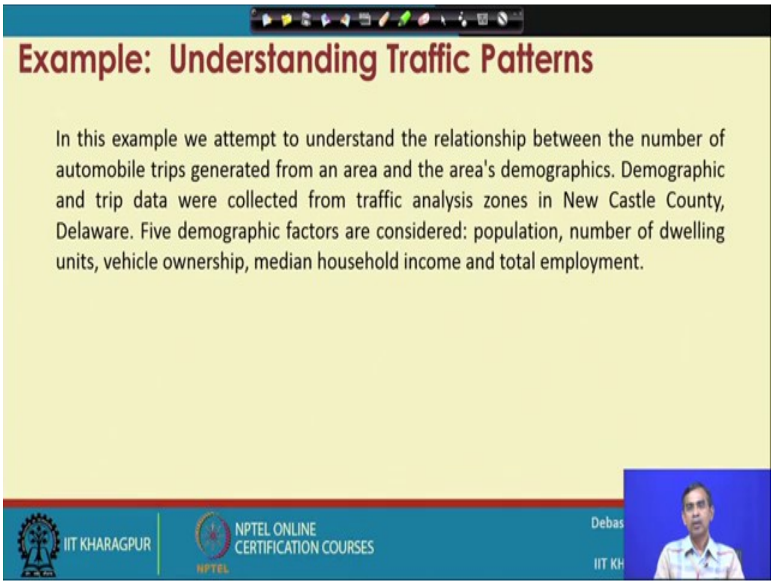
Now, so it is called the Matlab Fuzzy Toolbox. So, if you know Matlab, then you just in the Matlab if you type the command fuzzy, then this toolbox will be invoked. So, it basically have very good editor. It is called the Fuzzy Inferencing System editor, FIS editor. This editor basically in combination with four other editors which provides a very powerful environment to define and modify your fuzzy system or it is called the Fuzzy Inference System.

Now, you can recall at defining a fuzzy system is basically in terms of fuzzy sets, fuzzy rules, fuzzy membership functions, fuzzy inference rule and finally, the inference engine. So, the FIS can allow you to define all these things according to your own problem or application. Then it also has a very good tool set it is called a fuzzy controller.

So, this is basically a fuzzy toolbox is a block block in the fuzzy toolbox in the library called Simulink environment. This block allows FIS variable produced by the FIS editor, and then implements the many rules base system, and then controller that controller you can define either using Mamdani approach or using Takagi sugeno's approach.

So, these are the toolbox if you know the concept, and then toolbox are available to you, then we will be easily able to use the toolbox to solve your problem. Now solving your solving a problem means you have to decide the fuzzy membership functions for the different fuzzy element, then fuzzy rule base matrix, then fuzzy inferences all these things. So, they will allow you to enter all these things in a user friendly manner using graphical user interface in the matlab toolbox.

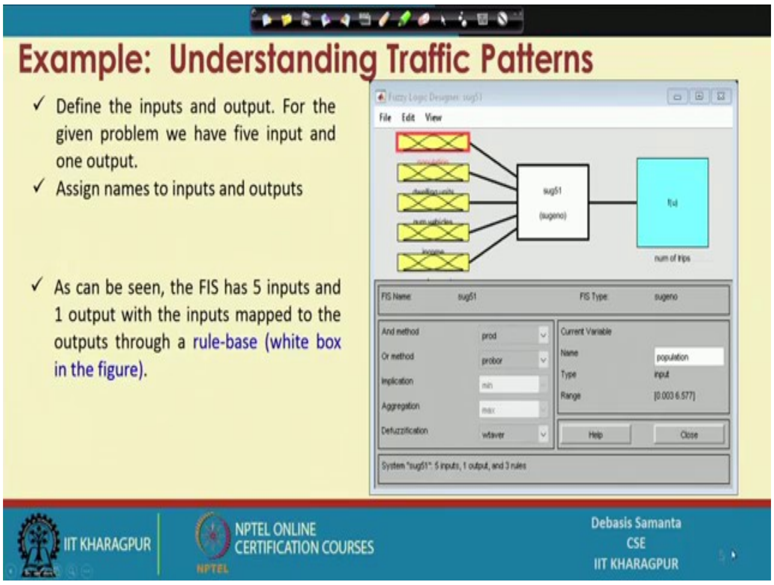
(Refer Slide Time: 05:18)



Now, so this is the toolbox it is called the matlab fuzzy toolbox. That is there in Matlab right. So, just it is very difficult to include the solution, because I want to give an introduction to all the toolboxes or all the tools that is available to solve the soft computing problems. So, it is one problem and this problem you can see it is basically traffic pattern recognition problem using the fuzzy system.

So, so you can decide what are the inputs to this system, and then all for all the inputs you have to fuzzify it and then for the all fuzzified inputs. Fuzzify means you have to decide the different membership function, and then all the membership functions once it is defined, then you can discuss the rule based system. That means, if then else rule that basically you have to decide that if this is happen you can recall what is the rule based system that we have discussed, and all those rule base you can define using these toolboxes.

(Refer Slide Time: 06:18)

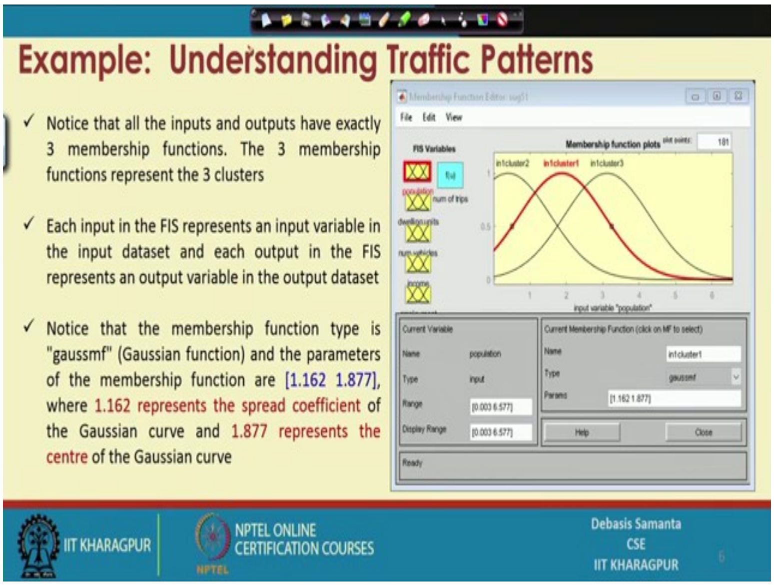


So, here is basically I mean view of the editors, where the different interfacing function can be there, and you can just simply using plotter, and plane. We can discuss a different membership function for the different fuzzy elements, and then the different the fuzzy rule base can be entered, and then we can decide the fuzzy systems or fuzzy controller.

And here the menu it is there which menu you can decide you to. Finally, adjust your membership function, and this menu also can give you the link of menus of the different membership function. You can select one. For example, you can select bell shape function, or trapezoidal function, or triangular membership function, and then different parameters in these membership functions by setting the different value here and then all the functions can be there.

So, basically this tool will allow you to decide or define all your fuzzy members, or fuzzy elements for your application. Now so, so this way you can enter every fuzzy members or fuzzy elements, and then the fuzzy membership functions, and the fuzzy rule base and others.

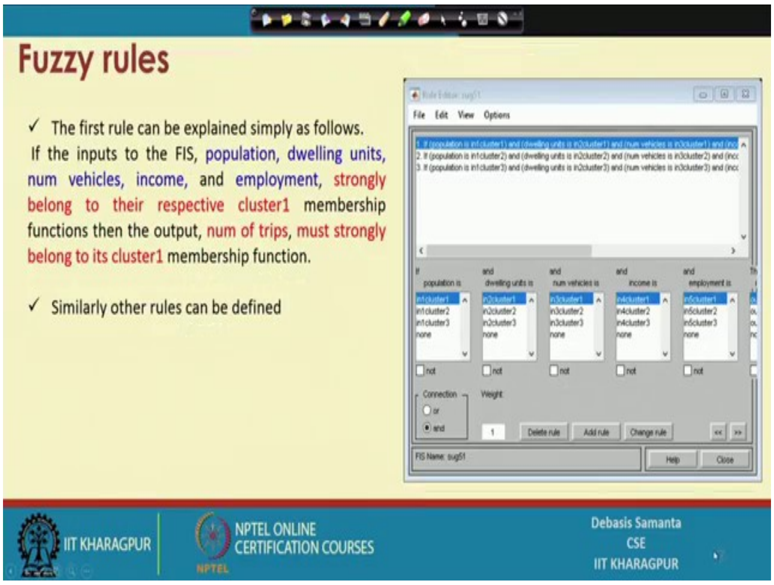
(Refer Slide Time: 07:23)



Now once it is there then you can also know exactly what is the output that the system can be a given to you it is in the form of fuzzy output. So, for example, for certain input the fuzzy output which can look like this one.

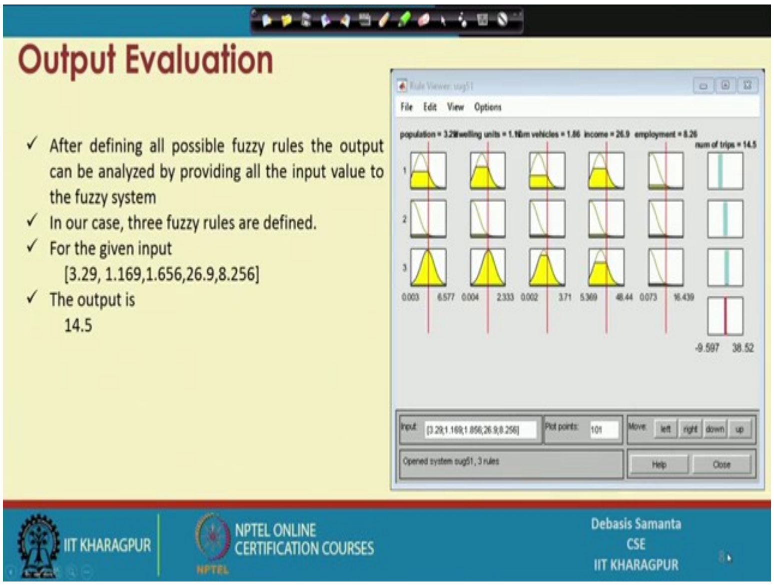
So, this fuzzy output again can be converted to the creeps output by using some fuzzification method. So, in the tool based method it will allow you to decide which fuzzification method that you want to follow, and then after your decision the toolbox will give the creeps output for the fuzzy values the defuzzified values. So, this is a tool that is there.

(Refer Slide Time: 08:12)



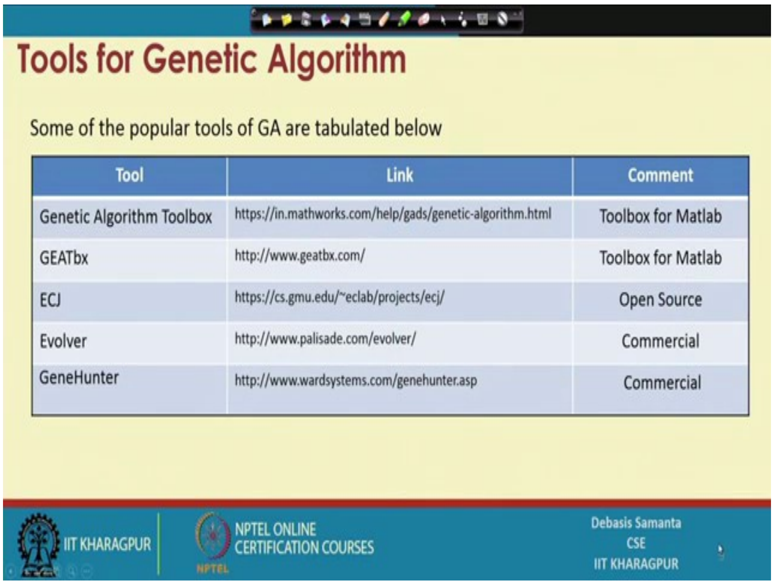
So, it is basically fuzzy tools that is there in the fuzzy toolbox system. Now, here also one example, how the fuzzy rule based can be entered into there. So, this basically allow a graphical user interface to enter the different rules are there.

(Refer Slide Time: 08:39)



So, you can apply it and then the fuzzy rule based system can be developed, and then fuzzy controller can be implemented. So, so this is the tool that is the there for the fuzzy logic controller in case of fuzzy tool base, and this basically shows how the output fuzzification method. It is basically how the output can be decided.

(Refer Slide Time: 08:48)

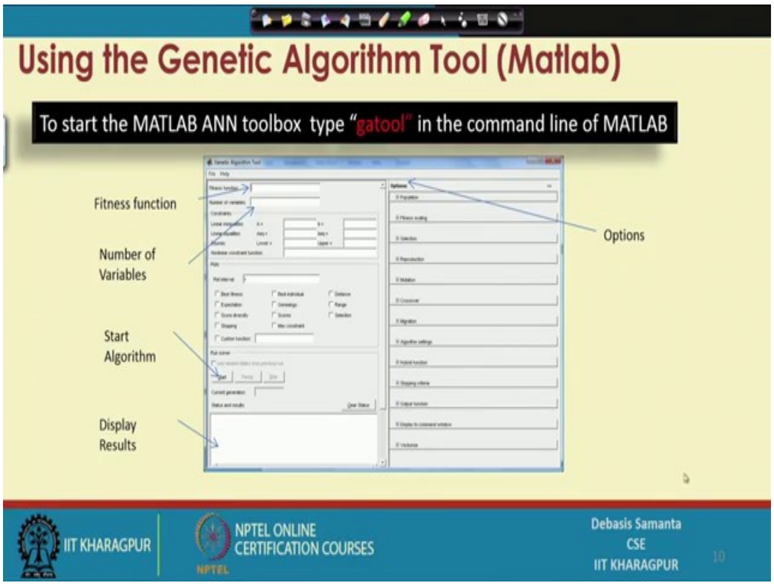


Now, so there is a tool and I have just given an idea about the glimpse of the idea about the fuzzy toolbox. That is there in Matlab other toolbox is likewise right. So, once you have the idea about that fuzzy concept then handling this toolbox will not take much time. But it is a matter of practice. So, basic idea of the practice is that you decide one program to be solved, and then decide the different elements that is there. Then different rules, and then the inference engine, and then you can allow the tools. Then all the steps that is there in the fuzzy computing can be carried out, and your problem will be solved.

Now, now next let us discuss about the tools for genetic algorithm like the open source tool. There is also an open source tool is available. This one is called the ECJ. This is the tool is developed by GMU. That is a good software or program repository a lot of programs will be there, and this is an as open source. Other than the open source, there is there are two commercial softwares for the genetic algorithm solving, they are called Evolver, and another is GeneHunter.

So, these are the two commercial source. Now, in the toolbox for Matlab again the two toolbox are there. One toolbox is called the GA toolbox. It is called a Genetic Algorithm toolbox that is there, and another is also this is for the multi objective optimization solving toolbox, GEATbox. So, it is basically GEATbox is there. So, we will discuss about genetic algorithm toolbox to solve the optimization problem a single objective optimization problem, using multi using Matlab toolbox. Now, again you have to consider one application. So that, you can practice the toolbox.

(Refer Slide Time: 10:56)

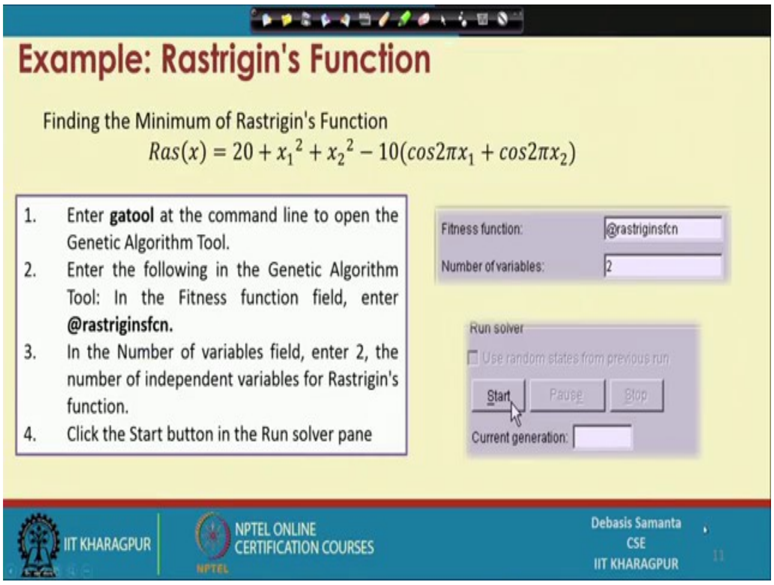


Now, it is basically a give an idea about the interface, or editor interface of the toolbox. This toolbox can be invoked by using the command GA tool in the Matlab commands, and then it basically defined many ideas for example. Using this interface, you can define what is the fitness function, what are the different constants are there, then what are the different, what is called the interval that you have to discussed. The different parameters that can be considered, and once you enter all the values that mean constant then objective functions, different parameters, the phenotype, genotype everything. Then once it is declared then you can start running the genetic algorithm.

And once the genetic algorithm runs it will give the output from each iteration, you can check it and then you can stop the running. If you see that output is not changing, that means, termination condition. And there are many other things that also can be set here which crossover technique. You can use which mutation can be, what are the different selection strategy that you can follow. You can take it there is a top down menu is there, if you select it. The different fitness assignment method will be there. Different selection techniques also will be there. Different crossover techniques are there. You can select some crossover technique use it, and then run your program to see the output.

So, this is very user friendly one toolbox which we can use it without knowing much details about how they are basically working. But, what they are supposed to do if you know and why they are doing like this if you, it is known to you then you can use this tool, and then solve your problem very easily, without any burden without any programming headache, and even without knowing any programming also. You can use this tool to solve your problem using genetic algorithm. So, this way this tool is very handy and very popular among the different students and researchers.

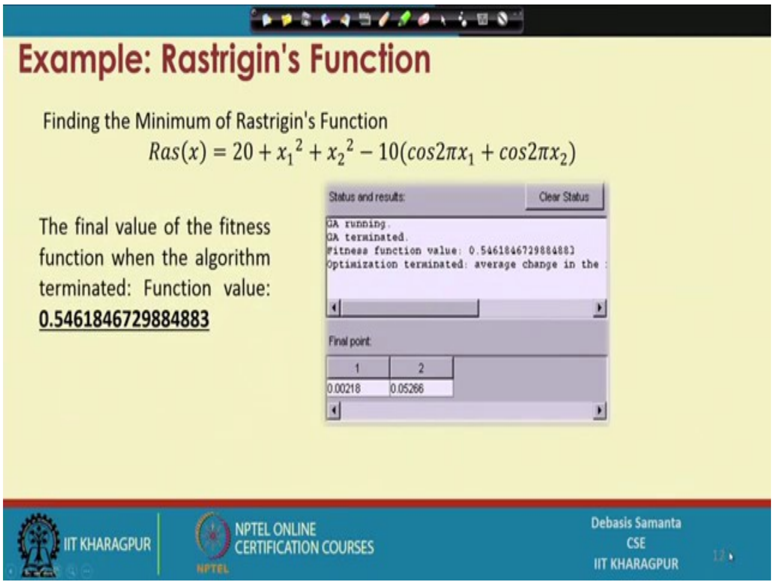
(Refer Slide Time: 13:06)



Now, here for example, you can try this GA toolbox to optimize this function this one. So, here basically what you have to do is that you have to enter this is the objective function, and then parameters that you have to discussed about x_1 and x_2 are the two parameters, and then in this case there is no constant mention. You can follow certain constant about that what is the range of the values of x_1 and x_2 from the link that is there in the things.

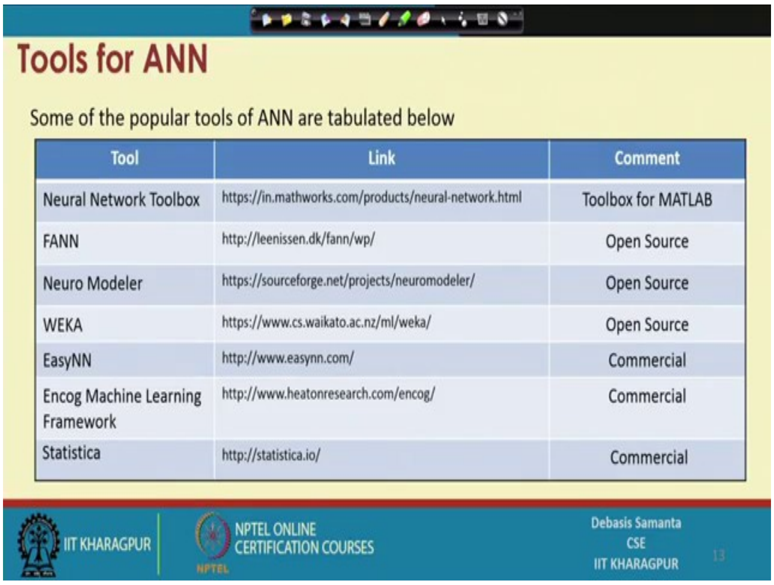
And then once the crossover technique whether it is a binary genetic algorithm or real value coded genetic algorithm all these things you specify. It will run this and then ultimately give the solution for this for this of for this problem. Now, if you run this particularly if you try with these tools, because Matlab tool is readily available in everywhere right. So, you can use this tool and then learn it.

(Refer Slide Time: 14:02)



It will give you finally, the output result like this a this output result for this; that means, this is the values of our x_1, and some other values of x_2 by this is a values of x_1 and values x_2 for each this gives you the minimum value, and the minimum value of this is this one. So, so far the accuracy is concerned it is very highly accurate, and then it will give it will solve your problem in a real time, and it is very effective and useful, and you can try solving the same problem once it is using binary GA, then using the real coded GA then using other GA techniques also, and then you can get the result and which GA technique gives a better result that you can use it and then finally, solve your problem. So, these are toolbox is basically there.

(Refer Slide Time: 14:50)

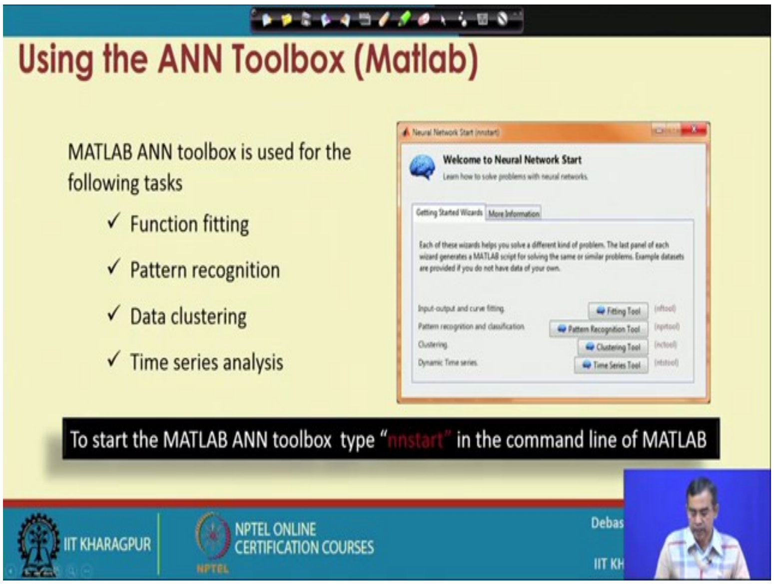


So, far genetic algorithm is concerned. Now we will quickly come to the ANN toolbox. There are many ANN toolbox is available. Some are open source. Some are the commercial toolbox. So, these are the open source tool box namely FANN. FANN, then Neuro Modeler, and then WEKA. WEKA is very one sophisticated and very powerful one toolbox to solve the neural network related problem, and there is also some commercial toolboxes which are here. These are the commercial toolbox, like EasyNN then Encog Machine Learning Framework, and another is Statistica.

So, these are the toolbox, and I have given the link from where all the tool box can be accessible. Those are the open source toolbox can be accessed from this link, and the commercial toolbox also can be obtained. I advise you to try or practice yourself with a WEKA tool which is very powerful.

Now, like the GA tools and then fuzzy logic tools, for the ANN also Matlab has a very good tool box. It is called the ANN tool box or Neural Network Toolbox. So, this is the link that you can use to access this tool box. If you have the Matlab, from the Matlab you can just give a command. You can give a command.

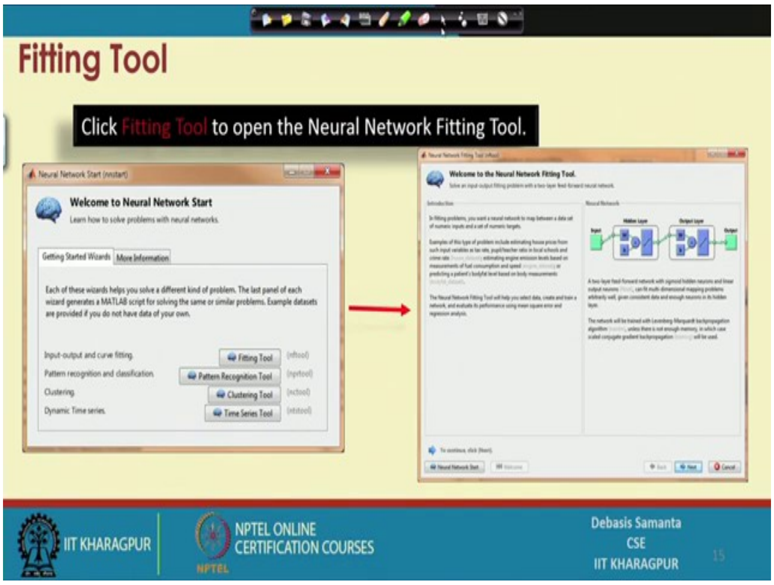
(Refer Slide Time: 16:19)



So, that you can it is NN start command. NN start command if you type it, then it will invoke the Neural Network Toolbox in Matlab, and using this toolbox lot of problems you can solve. I have mentioned the many problems. For example, it is basically program related to the regression analysis. These are the program related to the pattern recognition or classification, and these are the relation regression. This is the method for clustering technique, and this is basically time series analysis.

So, I have mentioned many problems, where the ANN can be used can be applied to solve all the problem. Now, this toolbox is also very similar to the other toolbox that we have discussed, in the context of this it is just like a simple user interface by which we can define.

(Refer Slide Time: 17:15)



The input size the output size, and we can give an input as the training data set to the net. So, it will take it and then finally, it will model it will learn the neural network, and for the learning neural network, again you can follow any technique that we have discussed either supervised or unsupervised or Hevian or Competitive learning or any other Task. So, toolbox have the all implementation of all the concept it is there. We have discussed for example, back proportion algorithm for as a Stephen descent method to learn to tell a neural network.

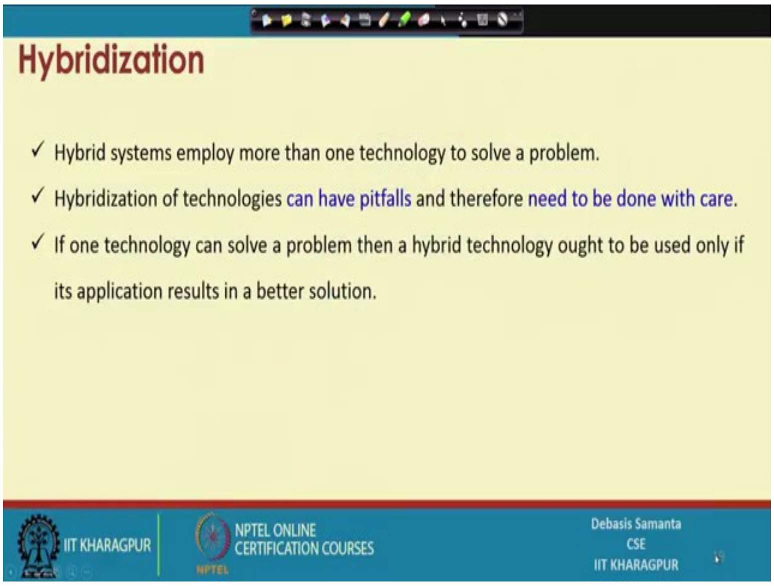
So, likewise back proportion there are many other training method, also known and then you can select from the toolbox apply it, and then it will solve for you. So, again like the other. There is no headache for the programmer. So, for the coding effort is concern coding is by the coding is behind the tool you can use these toolbox as the white box like; that means, give whatever the specification according to your own judgment give it to it.

And then system will take it and then system will solve the problem for you will get ultimately the final result that this is an neural network has been model, and this is the output. So, for any unknown data if you give this data to the model, it will give you the result like this one, just like a pattern recognition or classification or clustering this kind of problems are there. So, only one thing that is very much essential is that you have to know exactly what is your application, what is the specification of your problem and how you can use this problem, and then using this problem how you can solve it, and how the different things can be achieved.

So, this is the different tools related to the fuzzy logic, related to the genetic algorithm, related to neural network computing, we have discussed. So, it is just introduction and then ultimately it is it depends on your own practice, and then effort that you can spend to learn it more effectively, but for learning it requires how you have to decide some objective problems. So, if the problem is known to you, then you will try all these tools to solve your problem, and then you can have the idea about that these tools how it works to solve the problem.

Now, we will discuss about the concept of hybrid computing, we have discussed three computing paradigms mainly the fuzzy logic, genetic algorithm, and then artificial neural network. Now in case of hybrid computing, it is very interesting to know, whether all the computing that we have learn can be applied to solve a particular problem, or say suppose both fuzzy logic and genetic algorithm can be applied to solve problem, or say GA ANN or fuzzy ANN it is the concept and this concept is called a hybrid computing.

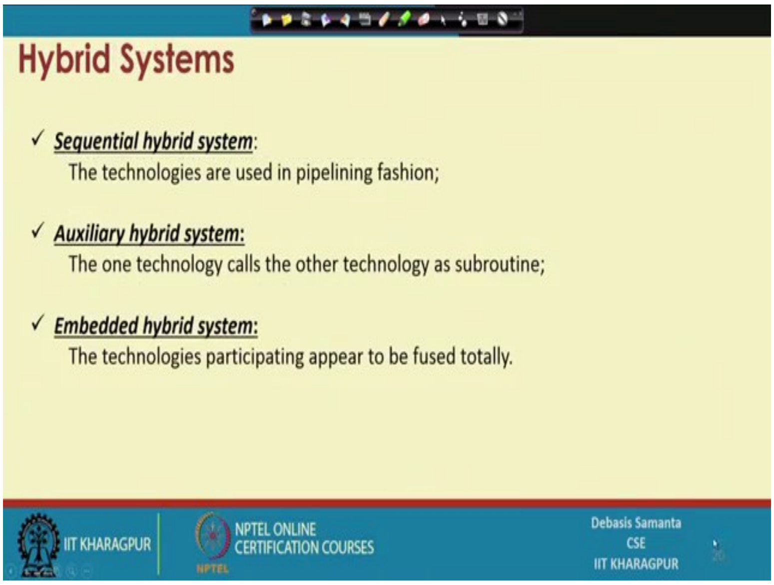
(Refer Slide Time: 20:19)



Now, for this hybrid computing so, here basic idea about is that you have to know exactly which program can be solved which computing better. For example, if you do not know precisely about the input, then you should try to solve this problem using fuzzy logic. If there is an optimization problem you can think about solving GA. For example, GA ANN can be clubbed together. So, ANN can give you the model parameters.

Now GA can help you to decide. What is the optimum number of model parameters that is required for a particular problem? So in that case it is NN followed by the GA is useful to solve your problem, and it is called the GA ANN techniques. So, like this GA ANN techniques, there is a GA FLL or GA FL ANN techniques are also there. So, we will quickly discuss about the different concept in this regard.

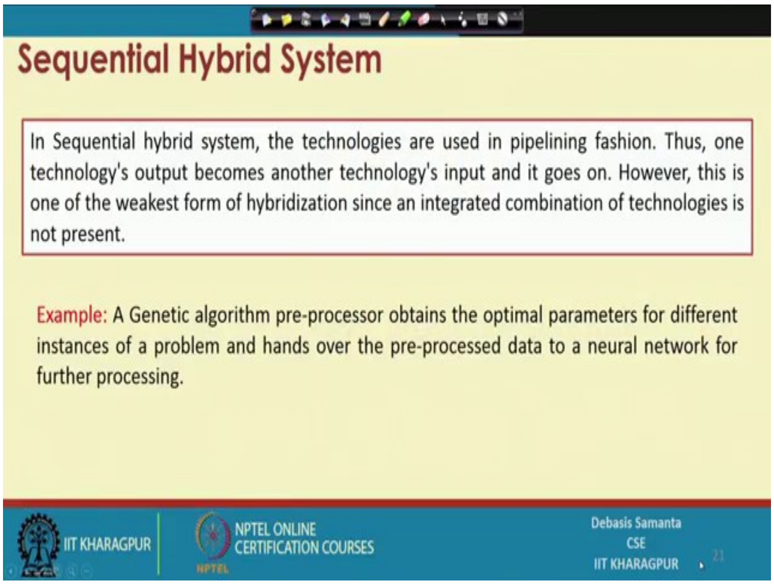
(Refer Slide Time: 21:18)



Now, any hybrid system which basically requires two or more that sub computing things and they can be classified into three broad category. One is called the sequential hybrid systems. In case of sequential hybrid system, one technique will be used then followed by the next technique. It is a pipeline fashion. On the other hand, auxiliary hybrid system is basically to solve one problem. We can follow say neural network, but neural network will call GA techniques as a subroutine.

So, it is called the auxiliary hybrid system, and embedded hybrid system is basically the different components of the problem can be solved with the different computing techniques like say GA, ANN and the ANN fuzzy logic. So, it is basically so you have to know only, and this kind of systems is basically useful for if the system is very large and complex. So that, the embedded hybrid system can be used to solve the problem.

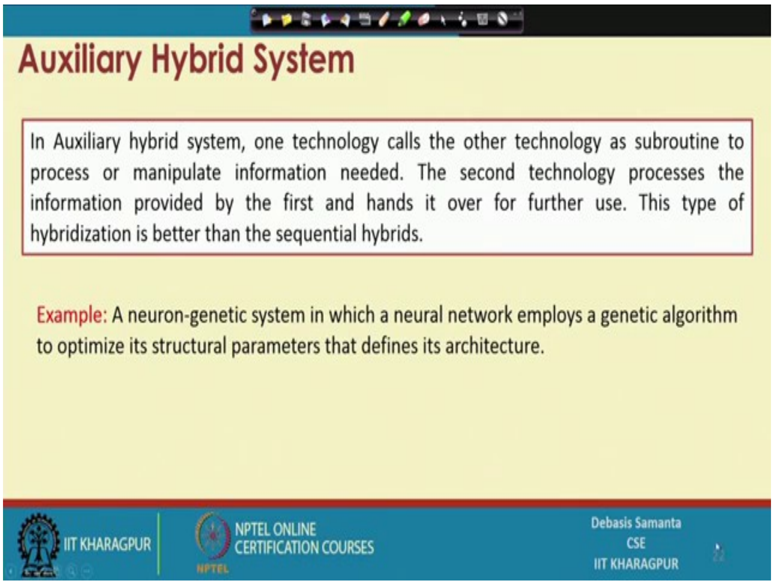
(Refer Slide Time: 22:31)



Now, sequential hybrid system that we have discussed about that here as, I told you the different computing are to be used in pipelining fashion. So, in other words, if there are different cascaded what is called the functions are there. So, it will basically take the one technology maybe say GA which will produce an output this becomes the input to the next stage and so on so on.

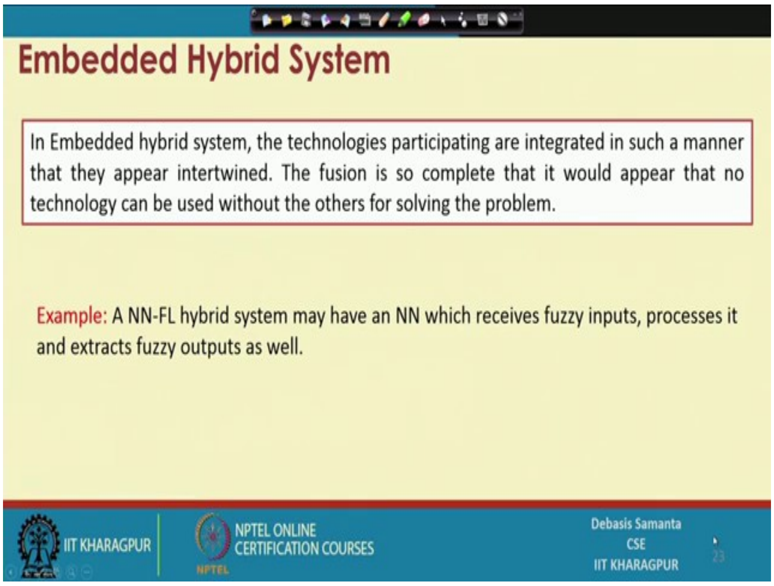
So, this is basically a sequential approach and gives rise to a sequential hybrid system. As an example, I can say genetic algorithm can be considered as a preprocessor which basically gives you the optimal parameters for different instance of a problem, and it basically give the preprocessed data to a neural network, and the neural network use it. So, the problem can be solved not only accurately, but it is also solved in a more faster way than any other method. So, it is most the quality as well as the speed can be enjoyed, if we use the hybrid system.

(Refer Slide Time: 23:45)



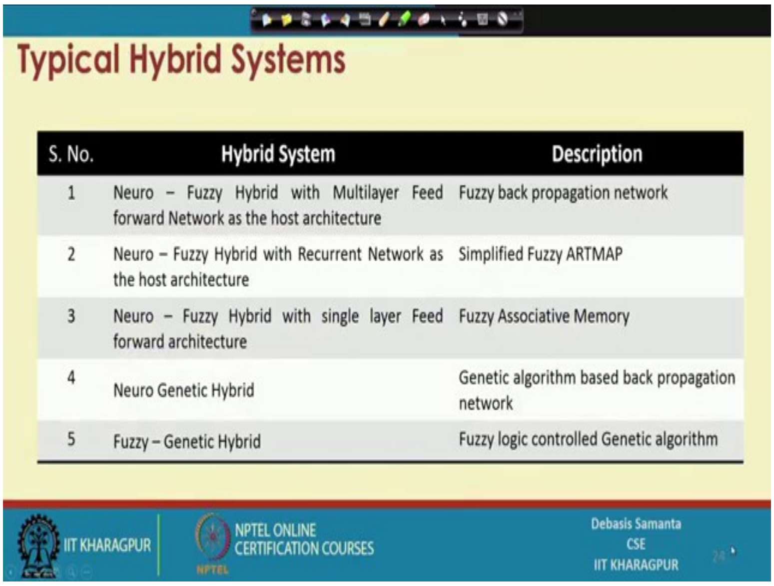
So, this is the sequential hybrid system, likewise the auxiliary hybrid system is basically the as I told you is basically one technology can be used as a subroutine, or it is a function to solve the other technology. So, it is for an example a neuro neural genetic system; that means ANN GA combination, in which neural network can be employed a genetic algorithm to optimize the different structural parameter, and then the optimum architecture can be obtained. So, this is the auxiliary hybrid system that can be considered as an hybrid system.

(Refer Slide Time: 24:25)



And then embedded hybrid system, as I told you there here the different technology can be used to solve the different parts of a very complex problem. For example, here neural network, and fuzzy logic, can be embedded together to solve where the ANN which receives the fuzzy input, and process it and it will extract the fuzzy output, and then finally, the result can be obtained.

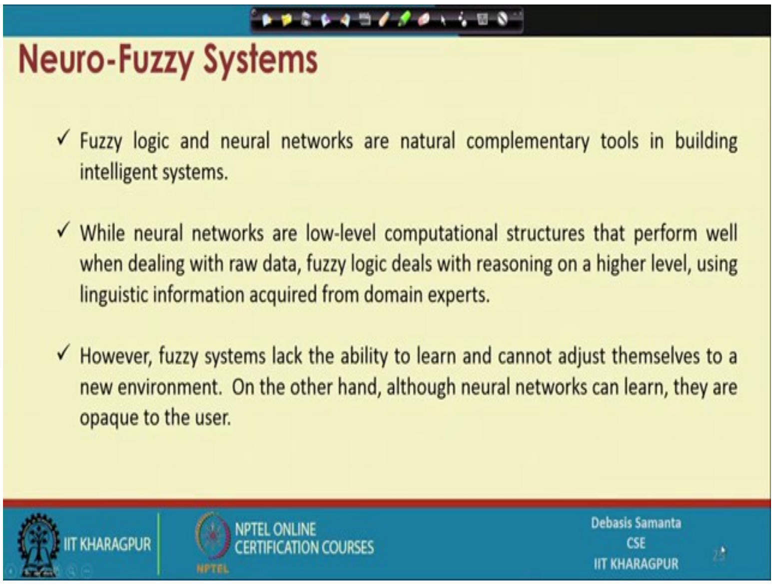
(Refer Slide Time: 24:57)



Now, as an illustration I can give an example ok, there are few hybrid system therefore, we can have it is also some tool box also available for the different fuzzy systems are there, one hybrid system is called neuro fuzzy hybrid with multi layer feed forward neural network as the host architecture, it basically used fuzzy back propagation network.

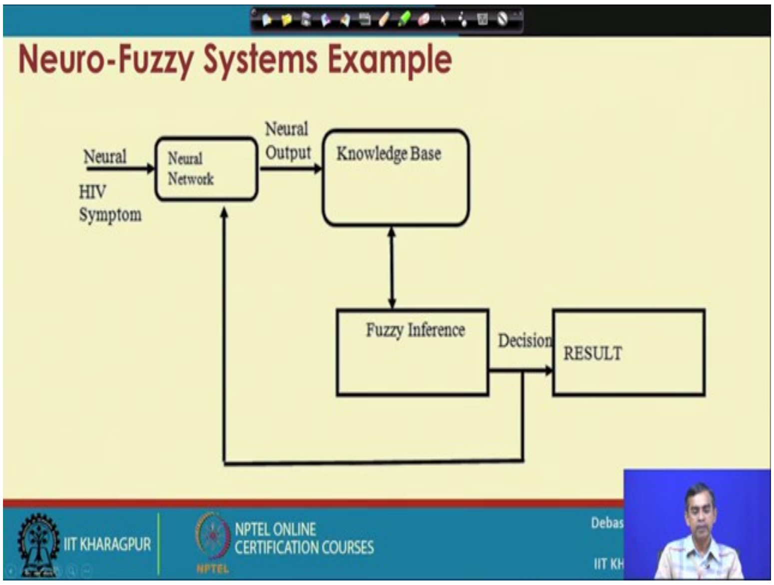
Likewise neuro fuzzy hybrid with recurrent network as the host architecture is basically called art map the simplified fuzzy some fuzzy problem, and then neuro fuzzy hybrid with single layer feed forward architecture is also known; it is called the fuzzy associative memory architecture or tools. The neuro genetic hybrid system is also known it is basically the genetic algorithm based back propagation network. Similarly, fuzzy genetic hybrid system is also known, here basically fuzzy logic control the genetic algorithm has been proposed. So, these are the different what is called a hybrid system is known at present, and it can be used to solve many problems for our problem solving a domain.

(Refer Slide Time: 26:10)



Now, so neuro fuzzy systems what we have discussed about it.

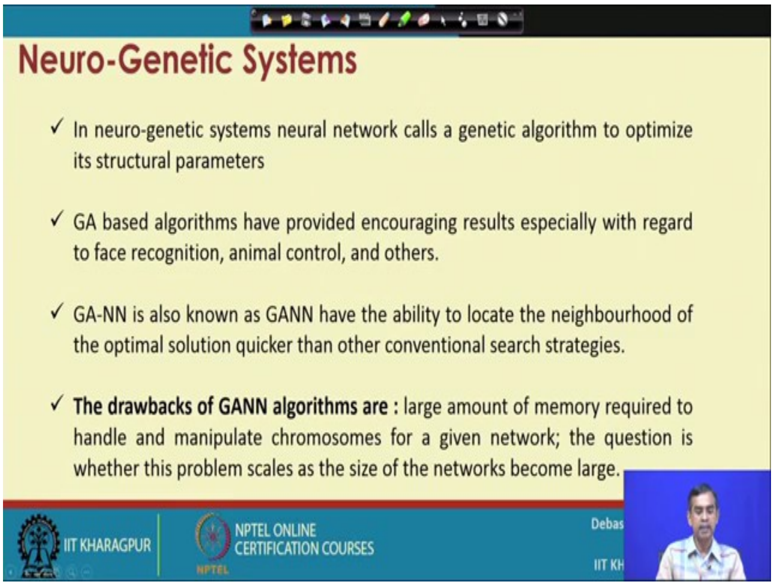
(Refer Slide Time: 26:12)



Now, here the idea about how the neuro fuzzy system works for you it basically take the input, as a training data, and this training data is basically for example, some disease symptoms, then it gives you the neural network, neural network will be trained, and the trained network will give output for an input, and this output will be used to develop the knowledge base.

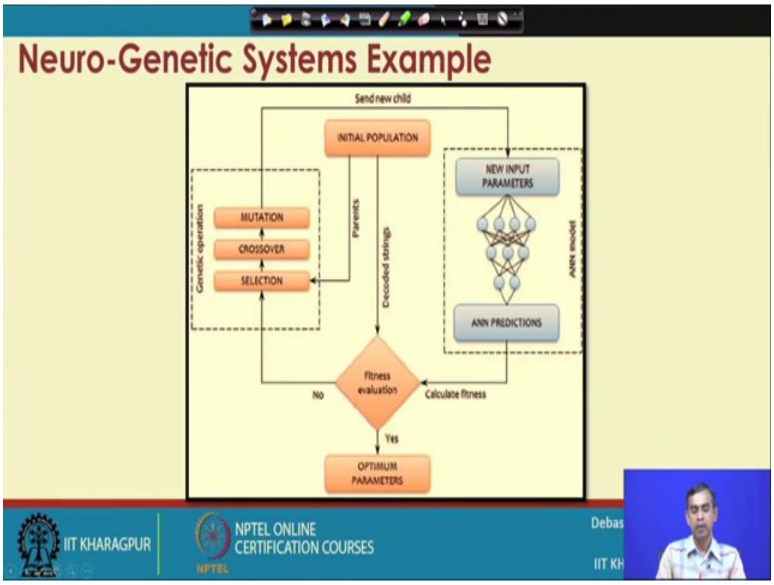
And using this knowledge base the fuzzy inference will be here, now here the neural output is basically in terms of fuzzy fuzzy input like, and then fuzzy inference give you the decision this is ultimately result, but it can be feedback to this one so, that it can be repeat, and the system can be fine tuned and finally, the hybrid system can be decided. So, this is the idea about that, how the fuzzy how the hybrid system works, and it is for the neuro fuzzy system that we have discussed.

(Refer Slide Time: 27:09)



Similarly, neuro genetic system also can be obtained, and here is idea about how the neuro genetic algorithm is there.

(Refer Slide Time: 27:12)

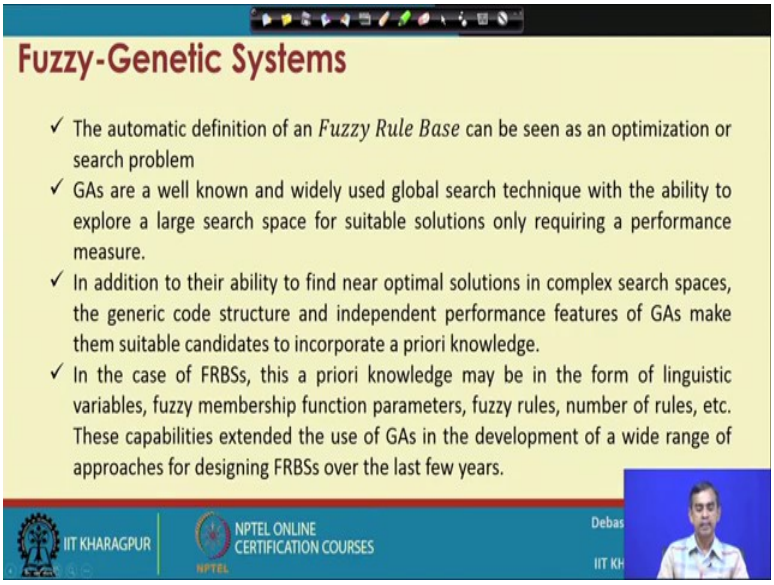


So, it basically the neuro neural network it is there, and all these things are basically embedded system that we are discussing about; that means, the problem can be solved using the different components in the different parts here.

Now, it is basically the genetic algorithm we approach, suppose GA and then NN can be clubbed together to solve the problem like, here the idea about is that so far the genetic algorithm is concerned, it will start with the initial population, and then so these are all populations are used to generate the new population.

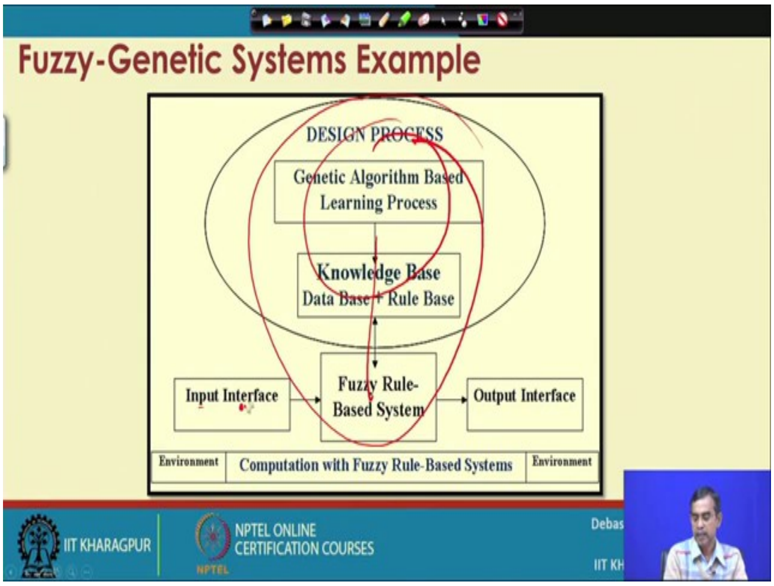
And this new population will go here, and then it train the network and that the network once trained it will check that fitness value, if the fitness value satisfy the optimum criteria it will give the result if not so, again GA then ANN. So, it is basically GANN one what is called the loop system, and that can be used to solve the problem. So, this is a concept of neuro genetic algorithm as the hybrid system to solve many problem.

(Refer Slide Time: 28:22)



Likewise, there is a fuzzy genetic system also can be considered here, basically the idea about fuzzy genetic neural system.

(Refer Slide Time: 28:26)



So, it is basically genetic algorithm based learning process it is called, and here the input inference will given fuzzy rule base system will be developed, but fuzzy rule base system will be develop in terms of the consolation of GA approaches which is an optimization of the number of rules that needs to be considered to solve your problem and finally, other output will be there.

So, in this concept there is basically GA, and then fuzzy are embedded together to solve for certain input to get certain output. So, it is basically the computation system using fuzzy genetic hybrid approach. So, we have discussed about the tools and applications, which we can consider to solve our problem, and finally, the most advanced concept of computing it is called the hybrid computing, where all fuzzy GA neural network can be clubbed together to solve your problem most effectively, and more accurately. So, with these things are you want to stop it here. I hope you have understood the basic concept. the course is in an introductory level. So, the introduction to the different concepts have been given, and you have enjoyed this class.

Thank you very much.