**Selected Topics in Decision Modeling**
**Prof. Biswajit Mahanty**
**Department of Industrial and Systems Engineering**
**Indian Institute of Technology, Kharagpur**

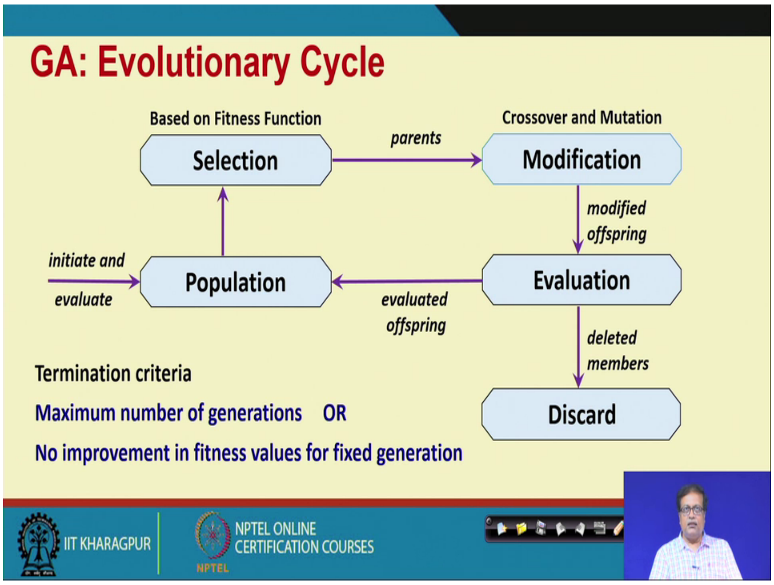
**Lecture - 33**
**Genetic Algorithm Process**

So, in our course on Selected Topics Decision Modeling, today we are in the 33rd lecture that is on genetic algorithm process. In our previous lecture, we have seen; what is genetic algorithm and we have understood that this is the first of the nature inspired algorithm. And the way the species, I mean as per the theory of Darwin moves from generations to generations. And as they move from generations to generations, they improve upon the fitness to the environment and through a process of say cross over and sometimes also through a process of mutation.

The similar thing is also adapted in case of optimization problems and we create you know the population of feasible solutions and there is a fitness function which is equivalent to the objective function and if you know we have some solutions which are more fit than others, then they have more chance of getting selected. And out of them, we have a cross over and a mutation process through which we create the new population for the next generation and why this method, as we move from generations to generations, we create finally, a set of feasible solutions with higher average fitness values.

And the best 1 should be closer to the optimal solution. So, that is the method which we are calling as genetic algorithm. Now, there are many variations to the genetic algorithm process and all this variations are essentially more of design concern which depends on the problem at hand and to some extent to the designer as well, right.

(Refer Slide Time: 02:38)

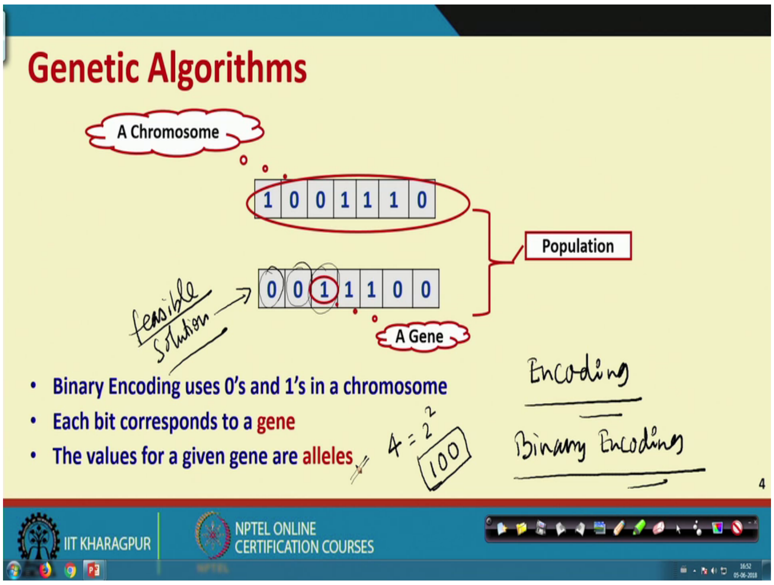



So, having understood that lets do a little recap and then through an example try to understand, what is really happening, I mean how to really carry out the genetic algorithm process and then in the next lecture, we shall see some examples.

So, this is the basic process as you can see that we start with you know population. So, we initiate a population this population is initiated through a random process right and as you initiate and then also evaluate the fitness of that population and after that we first activity, we do you know selection of parents based on the fitness function so; obviously, those parents, I mean those feasible solutions which are having higher fitness value has more chance of getting selected then the parents those who are selected they go through a process of modification sometimes called recombination of cross over and mutation.

Then after the cross over and mutation the modification that occurs then we have the modified offsprings and these modified offsprings are then evaluated, you know on the basis of what you call the fitness function again and may be sometimes, some members are deleted and some members are directly added from the parent population which is called elitism. And then that evaluated offspring you know that formulates the next generation population and the process goes on till we find the average fitness value is improving, is it alright.

(Refer Slide Time: 04:45)




So, that is the evolutionary cycle of genetic algorithm through which the process continues. The very first process out of these is a measure of that is encoding, is it alright; so an encoding process that is the very first process. So, when we do encoding, the encoding requires you know there are different methods of encoding 1 of the encoding is called the binary encoding. So, binary encoding is used sometimes. So, suppose I have a number, let us say 4, then 4 can be written in binary as equal to 2 2 to 3 power 2, is it alright?

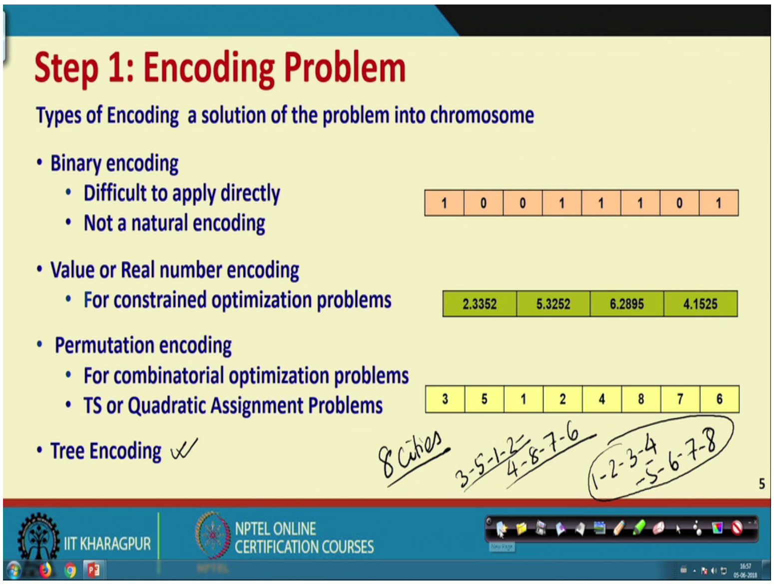
So, you know which is like 100. So, this 100 is you know 1 into 2 to the power 2; 0 into 2 to the power 1 and 0 into 2 to the power 0. So, that is how a binary numbers are written. So, you know you can represent some numbers through a chromosome and you know that is called the binary encoding having zeroes and ones.

So, essentially you know like every chromosome should represent a feasible solution, right. So, here population of 2 chromosomes are shown every chromosome is basically a feasible solution is a feasible solution to the problem sometimes that we shall see later, it may not be a feasible solution particularly where constraints are available that is not just an objective function, but we also have a; what you call a number of constraints then how to handle them sometimes we also may keep some infeasible solutions and see to it that their fitness values are pretty low.

Now, each bit corresponds to a gene. So, basically a chromosome is you know consisting of genes. So, you know here like this is a gene this is another gene this is another like this there are so many of them and the individual value for a gene is called an allele. So, there are alleles which are individual values of the genes and the bits are essentially like a gene which can have a 0 or 1 in case of binary encoding in case of real encoding it could be numbers between 0 to 9 and a, b, c, d, e, f, g, etcetera or you can also have sometimes hexadecimal or octal numbers, right.

So, that is about the encoding that is the first step. So, what we have to do if you if you really go back to our previous slide, then you can see that you know the initiate the population the population is to be initiated by defining a chromosome for a given solution. So, number of solution suppose you decide the length of population number of chromosomes in a population say 20. So, if you generate through a random means may be by using random numbers 20 chromosomes, then we have got a population to begin with, is it alright? So, that is how we start or initiate.

(Refer Slide Time: 08:45)




So, as we have initiated, then as I said the next is encoding and I have shown you a 1 kind of encoding that is called the binary encoding. Now, binary encoding is difficult to apply directly and not a very natural encoding, but sometimes as we shall see later in an example of you know where we have the knapsack problems the binary encoding could be a good option.

For some other problems, we may actually have value or real number encoding sometimes for constant optimization problems. So, here look here that each gene is actually a real number, is it alright, then for travelling salesman and similar such combinatorial optimization problems like quadratic assignment problems you know like quadratic assignment problem comes like in facility location problems where it is an assignment problem, but with a quadratic objective function. So, the method is you know go for what is known as a permutation encoding.

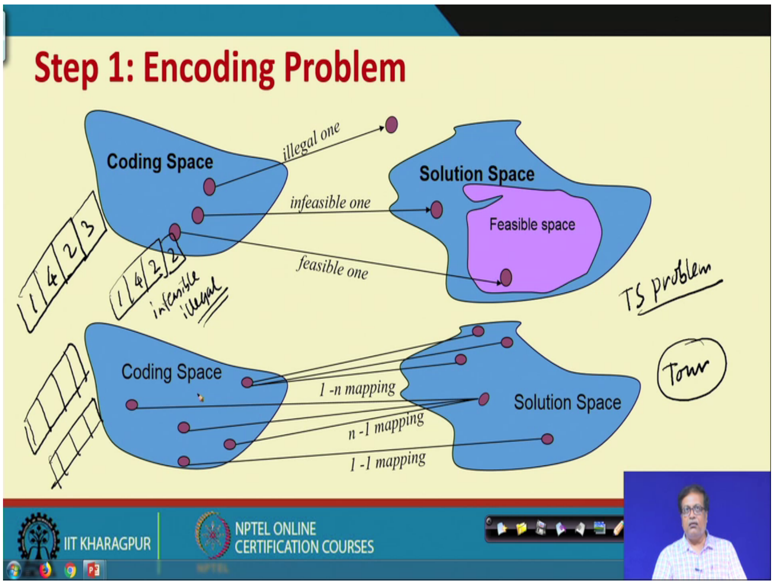
So, you see this permutation encoding you know this is an example. So, 3, 5, 1, 2, 4, 8, 7, 6; could be A gene. So, you see this suppose think of a travelling salesman problem, let us hope say there are 8 cities. So, there are 8 cities, there are 8 cities and a person has to go through all the cities exactly once before coming back to the city of origin. So, 1 such tool could be this 3, 5, 1, 2, 4, 8, 7, 6, no city should occur more than once. So, that is called permutation encoding

So, another gene that sorry another chromosome could be something like 1, 2, 3, 4, 5, 6, 7, 8, right. So, just going to the next city to the next city and after it coming back to one, but that we need not encode we can only encode this much to understand the particular you know the way the person moves from a city to the city.

So, 8 to 1; we will be considered in these particular case when we calculate the fitness function the taking those distances an example, I will show in the next lecture. So, that is how and then; obviously, there is also another kind of encoding which is known as the tree encoding, this is very interesting kind of encoding the tree encoding essentially is used when we are basically you know trying to find a formula a formula we want to discover a formula or a pattern then we can have tree encoding, right.

So, all these are different types of encoding obviously they are to be applicable for different kind of problems and given problem should be having a type of encoding, right. So, once we know what is an encoding.

(Refer Slide Time: 12:25)




We should also understand another very important thing that you know we have a coding space and we have a solution space the coding space and the solution space they are 2 different things, see the solution space is basically they are the actual solutions. So, assuming the same travelling salesman problem the TS problem, we are talking about in the solution space, we have a given tour a particular tour that the salesman has to move from a city to another city to this that say particular tour and that is in your solution space.

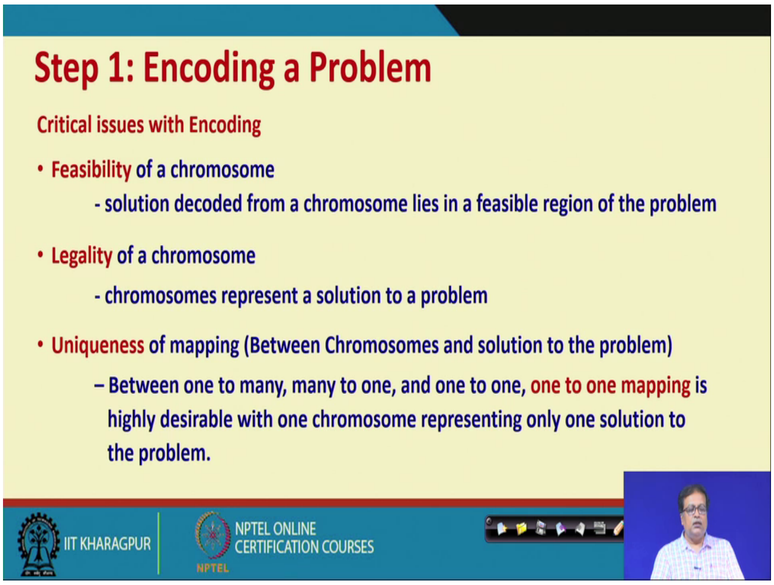
On the other hand the chromosome that you define you know for your or a number of such chromosome that you define for carrying out the problem that is your coding space. So, the coding space and the solution space you know as you can see there could be feasible, this is the feasible solution space there could be other solutions which are infeasible. So, there could be in the coding space we can have feasible solutions, we can have infeasible solutions for constraint problems and there could be some that could be illegal ones, right. So, we should be careful that we should not we should define coding space in first of all if we can define in such a manner. So, that only feasible solutions are available and nothing else

Say for example, suppose for the travelling salesman problem let us take a very simple 4 city problem. So, 1, 4, 2, 3 is a feasible solution, alright, but what about this let us say 1, 4, 2, 2; 1, 4, 2, 2; what about 1, 4, 2, 2, say 1, 4, 2, 2 is an infeasible not only infeasible, you can even call it illegal in a sense that you know, you just cannot really you know think of permutation going from city, 1 to city 4, then city 2 and then city 2 again, is it alright? So, it is not recall possible. So, that is what. So, if you say that I will have only you know; 1 number I mean no number should be repeated in the chromosome that is called permutation encoding it automatically ensures feasible solution will be created, is it alright?

So; that means, while you do encoding see to it that try to see that all the chromosomes come from the feasible space, is it alright and if absolutely not possible then accommodate some infeasible ones and take care of them by giving them, low fitness values, then there is mapping you see you should always try to see that there is a 1 2 1 mapping between the solution space and coding space; that means, every solution should be coded uniquely differently.

So, it should not be that 1 chromosome represents more than 1 solution should not be or similarly more 1 solution should not be coded you know differently 1 solution. So, there should be an 1 is to 1 mapping. So, these are some of the things that should be remembered while going for encoding.

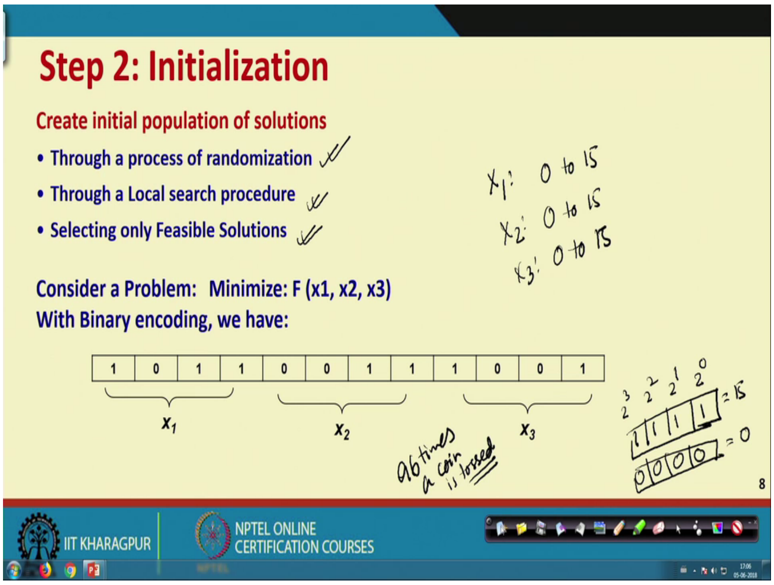
(Refer Slide Time: 16:29)




So, these are the issues the feasibility is an issue solution decoded from a chromosome lies in a feasible region of the problem then legality chromosome represents a solution to the problem and should not be something else and uniqueness of mapping; that means, there should be 1 to 1 mapping is highly desire desirable, right.

So, there should be 1 is to 1 mapping and if possible absolutely you know if nothing can be done then we should accommodate some infeasible solutions also with low fitness, but otherwise try to see the feasibility is maintained that is what is required for encoding of chromosomes in case of mapping the solution to the your coding set.

(Refer Slide Time: 17:26)




Now, comes the initialization the initialization means that we have to first create the population for the first generation the first generation population, we have to begin with somewhere. So, this is done through a process of randomization right a process of randomization through a local search procedure and selecting only feasible solutions right selecting of feasible solutions. So, supposing consider a problem minimize a function of X 1, X 2, X 3, is it alright and it is known that X 1, X 2, X 3, you know they are in a range which is between 0 to 15.

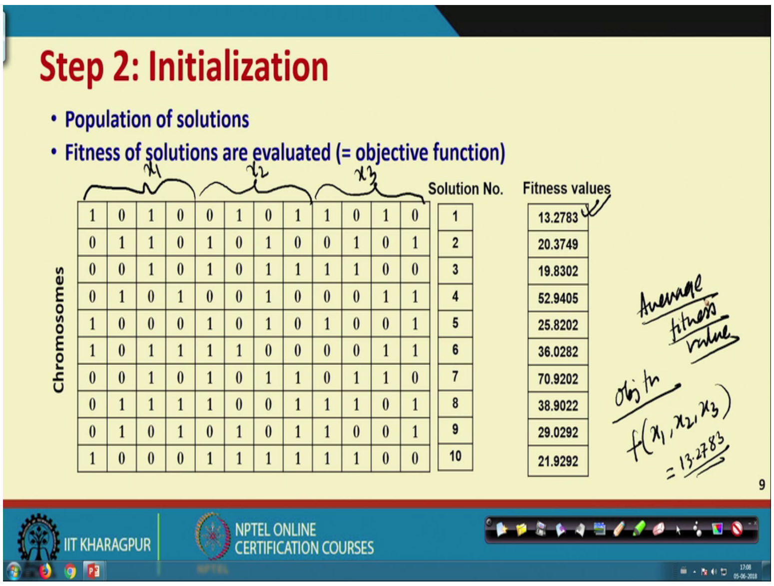
So, they are all hexadecimal numbers; that means, supposing it is known that X 1 varies from 0 to 15, X 2 varies from 0 to 15 and X 3 also varies from 0 to 15, is it alright. So, if this is you know what you what we can do what we can do not only 15 may be with a we are we using something like this; so it 0 to 15. So, supposing if we use 4 bits to represents one. So, you see the highest number is 1 1 1 1 and the lowest number is 0 0 0 0. So, these are all binary numbers. So, this is 2 to the power 3 2 to the power 2 two to the power 1 and 2 to the power 0. So, you see this will be 8 plus 4 plus 2 plus 1. So, this is actually 15 and this is equal to 0.

So, you see the any number in between will be between 0 and 15. So, through binary encoding suppose, we have a 12 digit string. So, this is our 12 digit string and you know we can represent a particular chromosome in this way that first 4 bits will be X 1 next 4 bits will be X 2 and next 4 bits will be X 3. So, supposing just arbitrarily assume suppose the population size is say 8 supposing. So, we have to then have something like you know tossing 12 into 8; 96 times supposing we carry out a tossing of the coin, is it alright?

So; that means, let us toss 96 times a coin. So, a coin is tossed 96 times, now supposing head is 1 and tail is 0. So, supposing first we get a head. So, I put 1 then a tail. So, 0 head head 1 1, then tail tail head head, then head tail tail head. So, that will be our first chromosome, is it alright? So, that is the randomization that we are talking about. So, you know like these if we do 12 times I get 1 chromosome and since the initial population size is let say 8. So, if I do you know 96 times, I can create an initial population of the solutions, right.

So, very simple process a population is created. So, once the population is created.

(Refer Slide Time: 21:32)

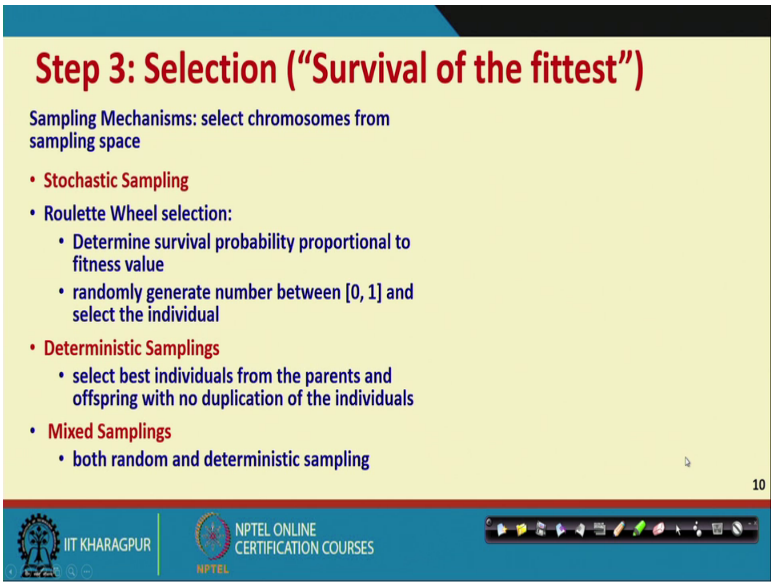


Then we go to the next stage that is. So, here you can see 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. So, here ten chromosomes are created. So, ten solutions are created. So, we had tossed something like 120 times. So, then assume the functional values, I have not given, but just assume these are the fitness values, right. So, you see once again please recall that the first 4 where x 1 next 4 where x 2 next 4 where x 3. So, second 1 was not correct. So, this was first 4, then this is second 4 this is third 4. So, this was x 1, this was x 2 this was x 3. So, these are the 3 variables.

Now obviously, the objective function objective function was a function of x 1 x 2 x 3 supposing when I put these value of x 1; that means, what is that value 2 to the power 3 8 plus 2; 10 here 4 plus 1, 5 and here 8 plus 2; 10 once again. So, when you put those values we get 13.2783. So, that we call as the fitness function value.

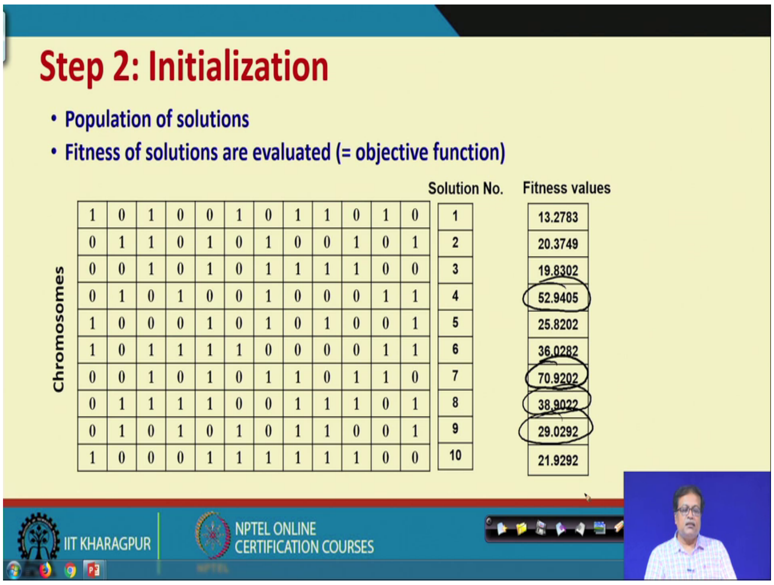
So, you can see that the various chromosomes has got various fitness values. So, 10 chromosomes of the initial population, they had a ten chromosome value. So, if I take what is known as an average of them, then I will get the average fitness, I will get the average fitness value; so, these average fitness value is going to help us in subsequent evaluation, right. So, that is how we say that how fit is this particular generation.

(Refer Slide Time: 24:02)




Now, after that we do a selection process. So, you see there are different kinds of selection 1 such selection is the roulette wheel selection. So, there is a sampling mechanism select chromosomes from the sampling space and roulette wheel selection is you know, I will explain right very soon the determine the survival probabilities proportional to the fitness value randomly generate numbers between 0 and 1 and select the individual. So, there could be deterministic sampling and there could be mixed sampling. So, select best individuals from the parents and offsprings with no duplication of the individuals and sometimes, we do a mix sampling with both random and deterministic sampling can be done.

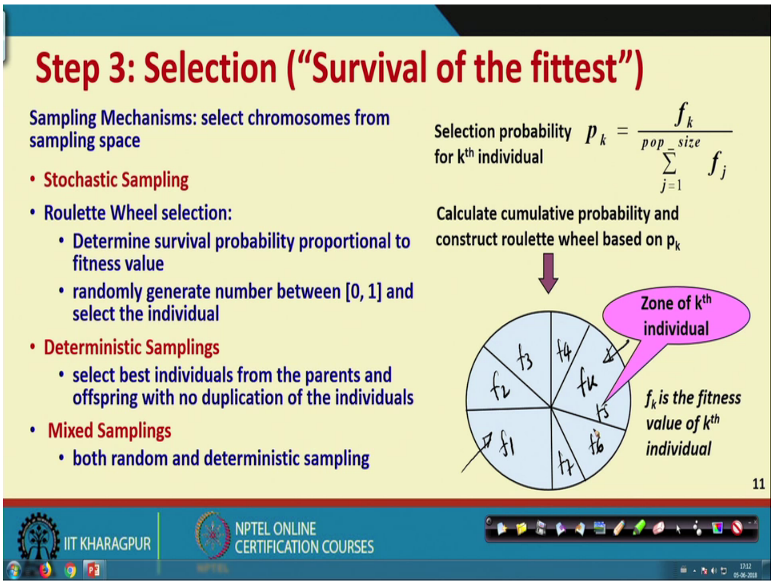
(Refer Slide Time: 24:55)




So, you see a deterministic sampling example could be here which ones have very high fitness values look here this is a very high fitness value. So, this is a very high fitness value, then this is also very high fitness value, then this 38 is also high and this 29; suppose we select this 4 directly, then what is that is we can call as deterministic sampling is, alright.

But supposing we give some randomization process and then I select then that kind of process can be called as a probabilistic sampling method, right. So, there could be these deterministic versus random or probabilistic sampling, sometimes, we can have a mixed sampling also which is a mix of the 2, right.

(Refer Slide Time: 25:51)



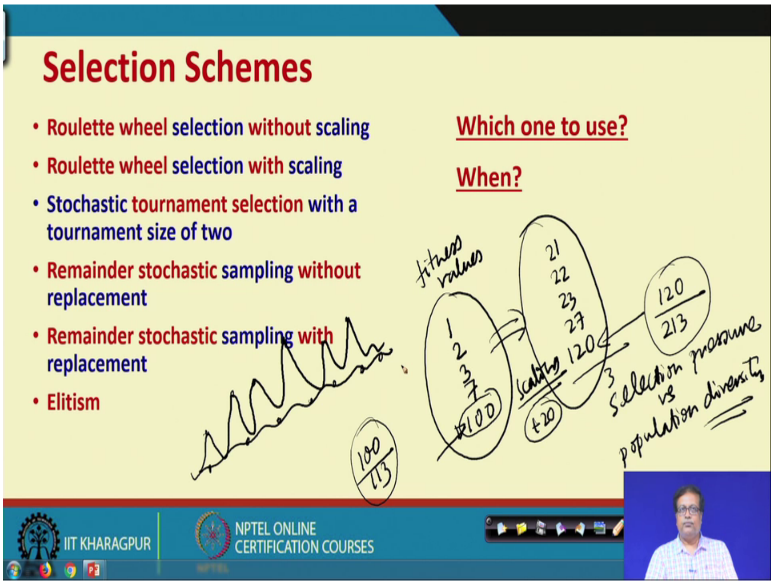

So, take this example of a sampling mechanism. So, select chromosome from sampling space stochastic sampling roulette wheel selection determine survival probability proportional to the fitness value and already explained. So, here what we do essentially you know we create the roulette wheel in this manner, is it alright?

So, you know we can take this as the zone of k th individual fk is the fitness value. So, this is like fk. So, supposing this is f 1, f 2, f 3, f 4. So, these are the fitness values. So, this is the k th 1 which is a 5 by the way, then f 6 and f 7. So, supposing I have 7 strings in our case we have 10. So, we do that and then you know by creating you know these kind of what you call circle, then when we rotate, this you know as we rotate, what we get is the is the result that result would you know supposing the chances of the f 1 getting selected will be or chances of fk getting selected can be given as fk by sum over j equal to 1 to pop size fj. So, sum of all the fitness and fk by that. So, that will be the probability of these fk th portion getting selected, is it alright, that is the roulette wheel selection, is it alright.

So, suppose this particular chromosome comes up. So, we select it supposing next time the f 1 comes, then the first chromosome will be selected, is it alright. So, basically selecting for what selecting for recombination what are some recombinations the crossover and the mutation is it, alright. So, there are crossover and there are mutations.

So, these are different kind of these things examples; so, right. So, having known this that how to how to do selections now we know that there are different types of.

(Refer Slide Time: 28:29)




Different types of selection for example, we have the roulette wheel selection without scaling roulette wheel selection with scaling stochastic tournament selection with a tournament size of 2, then remainder stochastic sampling without replacement remainder stochastic sampling with replacement and elitism.

So, you see to give a basic idea that what is the scaling method the scaling essentially is that supposing I have some numbers right. So, the numbers are such that you know they are you know say just; let us give example of the fitness values supposing the fitness values are 1, 2, 3, 7 and 100. So, if these are my fitness values and then if I do a roulette wheel selection, you know you will find the chances of these 1 getting selected will be very very high, is it alright.

Whereas, if I you know try to scale these say I want to scale it by simply adding 20 to each of them. So, what will be the new fitness values there will be 21, 22, 23, 27 and 120. So, this is the process of scaling. So, what is the scaling I have used plus 20; so you see the chance of these getting selected was 100 by the 107, 10, 12, 13, 100 by 13, 100 and here chance of this getting selected is 120 plus 113 and another 100. So, 213, is it alright. So, if you add them 3, 6, 3, 13, 1, 3, 5, 7, 9; 21.

So, you see 100 by 113 is very near to 1 whereas, 120 by 213 is not so much near to 1. So, what has happened the chance of these getting selected has reduced with scaling in other words the population diversity has gone up, why we need a higher population diversity, see we have a solution space suppose this is my solution space. Now there could be local and there could be global optimus. Now suppose this is my global optima and these are my local optima. So, if I have high population diversity, then I have representations of points all over the solution space otherwise, if a particular chromosome is selected all the time I have, what is known as you know more chance of the solution getting locally focused right with a narrow focus and chances of the local solution to be resulted.

So, that is why it is important that we should have a balance between selection pressure which is you know fitter chromosome should be selected versus population diversity which means the chromosome should be selected from all over the solution space right. So, I stop here and we shall continue this in our next lecture also.

Thank you very much.