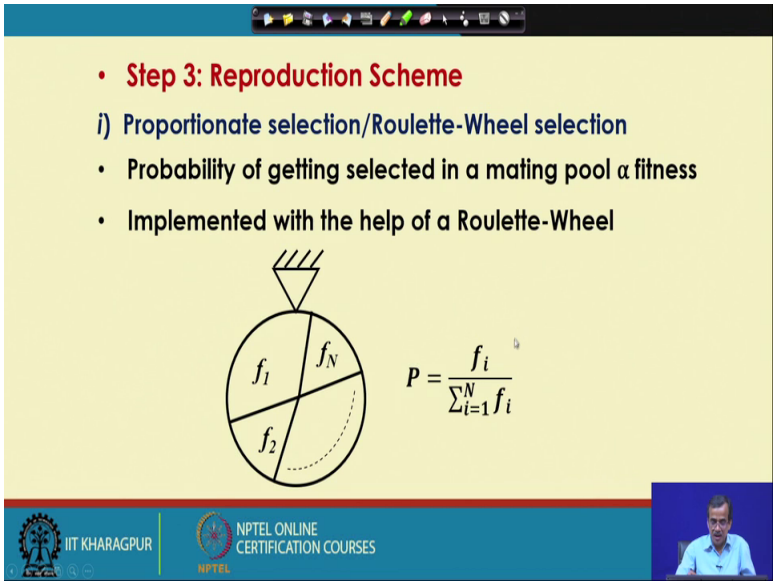
Traditional and Non-Traditional Optimization Tools
Prof. D. K. Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 05
Binary – Coded Generic Algorithm (BCGA) (Contd.)

We have got the fitness information for the whole population of solution selected at random initially. Now as I told that has the no as there is no guarantee that all the solutions will be equally good. So, what I do is we try to use one operator called reproduction scheme just to find out or just to select the good solution, which inform the mating pool. Now step 3, I am just going to discuss the reproduction schemers.

(Refer Slide Time: 00:41)



Now, if you see the literature we have got a few reproduction schemes, now out of all the schemes the oldest 1 is the Proportionate selection or the Roulette-Wheel selection.

Now, let us see the principle of this proportionate selection or the roulette wheel selection. Now here the probability of being selected in a mating pool is proportional to the fitness. So, the higher the value of a particular solution the more will be the probability of being selected in the mating pool. Now this particular the roulette wheel selection or the proportionate selection is implemented with the help of one roulette wheel, now I am going to discuss the principle of this particular the roulette wheel, now here we have got a fixed pointer and we have got these particular roulette wheel on the

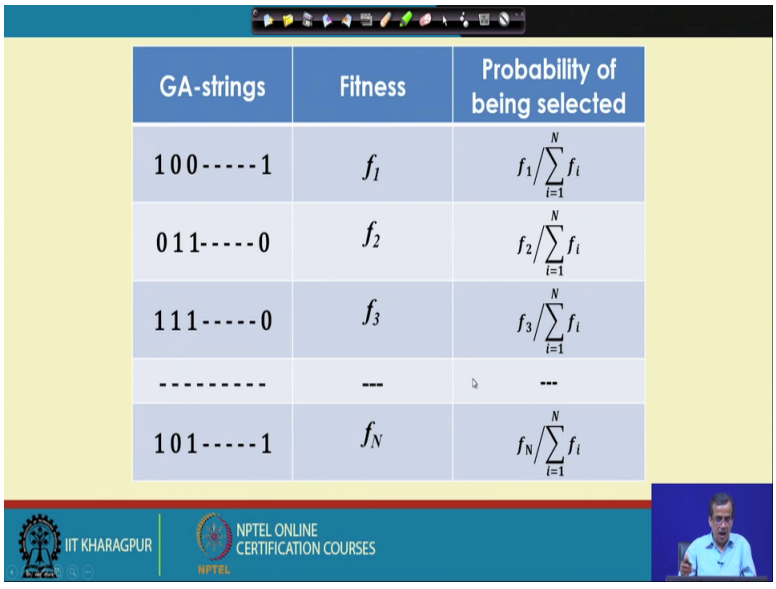
top surface of the roulette wheel there will be a few marking and the marking of this particular f 1 f 2 up to f n these are all the fitness function values.

Now, the area covered by a particular the GA string that is equivalent to it is fitness value. So, the higher the fitness value the more the area will be occupied by that particular the string on this the top surface of the roulette wheel. Now the way it works the pointer is kept fixed the roulette wheel is the rotated either clockwise or anticlockwise.

Now once it is rotated after sometime it is going to stop and if it is going to stop here the way of sonnet on this particular figure. So, f 1 is going to be selected for the mating pool. Now similarly the f N can also come in front of this pointer the fixed pointer in that case the N th string will be selected for the mating pool and so on.

Now, here the probability of selection is nothing, but the individual fitness divided by the sum of all the fitness values for the whole population. So, the probability that the i th solution will be selected in the mating pool is nothing, but f i divided by the summation f i i varies from 1 to capital N. So, n is nothing, but the population size.

(Refer Slide Time: 03:58)



Now, let us concentrate on the GA population. So, this is the initial population which I have got and that is created at random for each of the GA string I have got the fitness values. Now I can find out what is the probability of being selected in the mating pool

for each of this particular the GA string and once I have got the probability values for the whole population the sum of probability values will be equal to 1.

(Refer Slide Time: 04:32)

Now, here I am just going to put 1 note the GA-search depends on the factor like population diversity that is nothing, but exploration and selection pressure that is nothing, but exploitation. Now to discuss this further let me let me try to take 1 example now let me consider say dr x has joined say 1 institute as a faculty and he wants to go to the top he wants to reach that particular top position and how he will have to make all such planning of the activities.

Now, if you wants to concentrate on many various then population diversity will be more and if the population diversity is more he may not be able to concentrate on a particular area in depth, and consequently he will not be able to carry out search carry out research in that particular direction, he will not be able to produce good research resolves in that particular in a particular direction and consequently he may not get promotion in his career and there could be stagnation. So, if the population diversity is more that is exploration is more there is chance of stagnation.

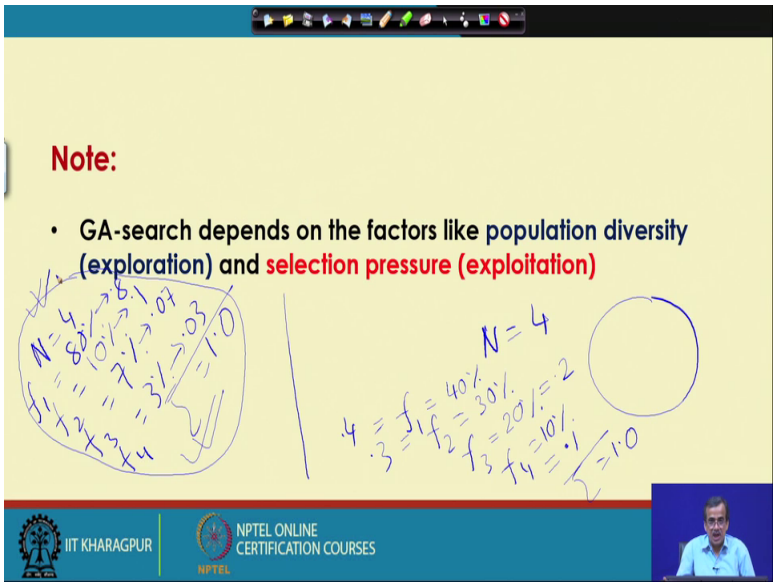
Now, on the other hand if the section pressure is too much there is a possibility he will be concentrating more on a particular area or 2 areas he will be able to go to the depth he will be able to produce good quality papers in large in number and there is a possibility that he will be getting the promotion with early; that means, to reach that particular level

might be he is going to exploit others. So, if the selection pressure is more the chance of exploitation is more, but there is a chance of premature convergence.

So, this particular fact has been copied in the artificial way in genetic algorithm also, now if you want to ensure a very efficient GA search there should be a proper balance between population diversity and selection pressure. So, that there should not be any premature convergence at the same time stagnation is also not welcome.

Now, how to how to set the premature convergence or how to remove that problem of premature convergence, now here actually what we do is if you see this particular the roulette wheel selection. So, there is a chance that there will be some sort of the chance of premature convergence. So, that particular problem is there now let me take a very simple example.

(Refer Slide Time: 07:59)



Some numerical example let me take supposing that I have got one population GA population of size say N equals to 4 and if I just consider one roulette wheel here. Supposing that for the first solution that is f 1 whose fitness is f 1 is say 40 percent of the total fitness, f 2 is say 30 percent of the total fitness. So, 40 plus 30 70, then f 3 is say 20 percent there is 90 and f 4 is the remaining 10 percent that is the 100 percent. Now if this is the situation what will happen is it is probability will be 0.4 it is probability of being selected is 0.3 it is probability of selection is 0.2 it is probability of selection is 0.1.

Now, this is one scenario I am just going to consider another scenario where same N equals to 4, but fortunately or unfortunately f 1 is say eighty percent f 2 is nothing, but say 10 percent, f 3 is nothing, but the 7 percentage and f 4 is nothing, but the 3 percent now here the probability of being selected is 0.8 here it is 0.1 here it is 0.0 7 here it is 0.0 3 and the sum of the total probability is equal to 1.0 here also the sum of the total probability will be equal to 1.0.

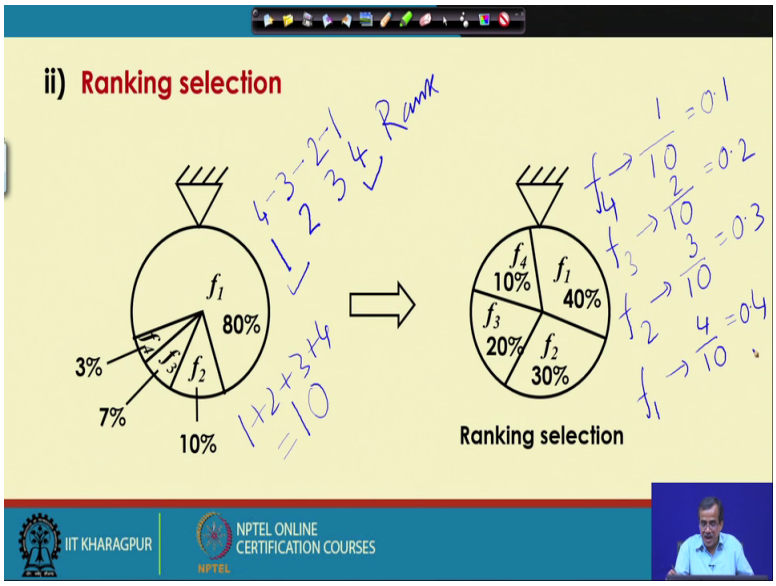
Now, let us compare these 2 scenarios, now here in this particular case the probability that the first thing will be selected is 0.8 now if this is the situation and if I use the proportionate selection or the roulette wheel selection. So, 80 percent of the total population will be occupied by the first string in the mating pool now 80 percent of 4 is 3.2. So, out of 4 strings might be the 3 positions will be occupied by the first string.

So, here the change of premature convergence will be more diversification will be less, but if I see this particular scenario here once again the population size is 4, but here the scenario is slightly different for the first string the probability is 0.4 second string probability is 0.3 third string is 0.2 4th it is 0.1, but here the chance of premature convergence will be less, but here the chance of premature convergence will be more.

So, if I use proportionate selection for this particular problem. So, I am going to face this particular the premature convergent the chance of premature convergence and less diversification, which is not the situation here for this type of problem. So, the proportionate selection all the roulette wheel selection may work in a better sense, but it may not work here because there will be more chance of premature on budgets just to summarize whatever I discuss like, if I use proportionate selection or the roulette wheel selection there will be a chance of premature convergence.

Now, to overcome that so what you do is we take the help of another proportionate scheme that is called the ranking scale now in ranking selection.

(Refer Slide Time: 12:03)



What we do is once again we take the help of the proportionate selection, but that particular proportionate selection will not be in terms of the fitness information not in terms of the probability value the way we calculated based on the fitness information instead it will be based on the ranking of the solutions in the population.

Now, let us see how does it work now this is the typical scenario which I discussed now here see for this particular f 1 there is 80 percentage chance that it will be copied in the mating pool and it will create the premature convergent just to overcome that we take the help of this type of ranking selection. Now what is do is now if I just concentrate here I can find out in terms of fitness like f 1 is having 80 percent, f 2 is 10 percent, f 3 is 7 percent, f 4 is 3 percent.

Now in the ascending order if I just hot the now this particular GA string the lowest fitness f 4 is the lowest fitness. So, the fourth solution will come fast then comes your next is the f 3. So, 3 will come then next is 2. So, 2 will come after that the first one will come. So, in this is the sorting in terms of your the fitness now if this is the sorting in terms of fitness what I do is we try to assign some rank the string which is having the lowest fitness that is assigned rank 1.

The string which is having the next higher is assign rank 2, next is rank 3, next is rank 4; that means, the string which is having the lowest fitness is assigned rank 1 and the string which is having the highest fitness is res assign rank 4 and now what I do is we add the

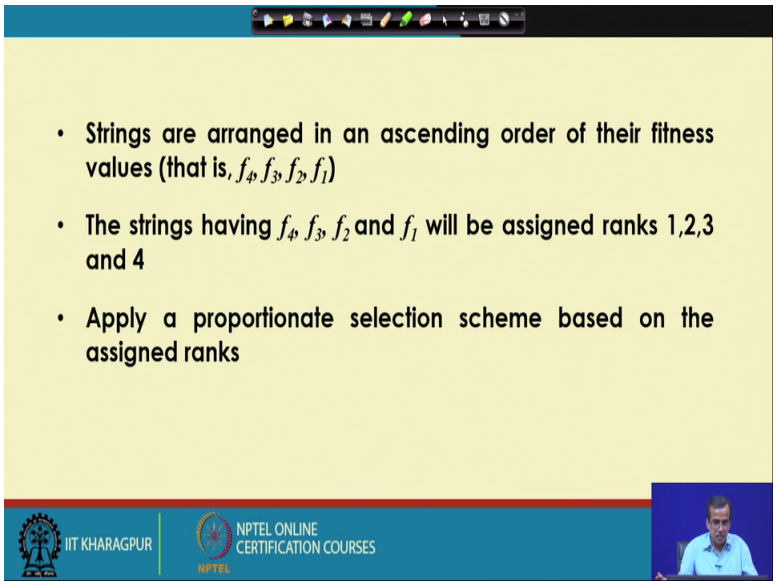
rank values like 1 plus 2, plus 3, pus 4. So, this is nothing, but 4 plus 3 7 plus 2 9 plus 1 10. So, this is 10 this is the summation of the rank values now here we try to calculate in ranking selection the probability values in a slightly different way that is for a f 4, which is having rank 1 will be getting the new probability value that is 1 divided by 10 that is 0.1.

According to the ranking selection then f 3 is having the rank 2 this divided by the sum of the rank values. So, this is the 0.2. Similarly your f of f 2 so this is the rank 3 divided by 10 that is 0.3 then comes your f 1 that is 4 divided by 10 that is your 0.4 now you see the probability value for f 1 previously it was 0.8. Now it is 0.4 for f 2 previously it was 0.1 and now it is 0.3 f 3 previously it was 0.0 7 and now it is 0.2 and f 4 previously it was 0.03 and now it is 0.1, now here if you concentrate and if you compare these 2 methods in terms of the probability values of only f 1.

Here it was 80 percent 0.8 and here it is 0.4. So, there is a chance that I am going to give chance to other solution to come forward towards the mating pool. So, here the diversification will be better compared to this and if you follow and if you give more chance for diversification there is a possibility there will be less chance for premature convergent. So, in ranking selection the chance of premature convergence will be less whereas, in case of proportionate selection that is your the roulette wheel selection the chance of premature convergence will be more. So, this is the advantage of this particular the ranking selection.

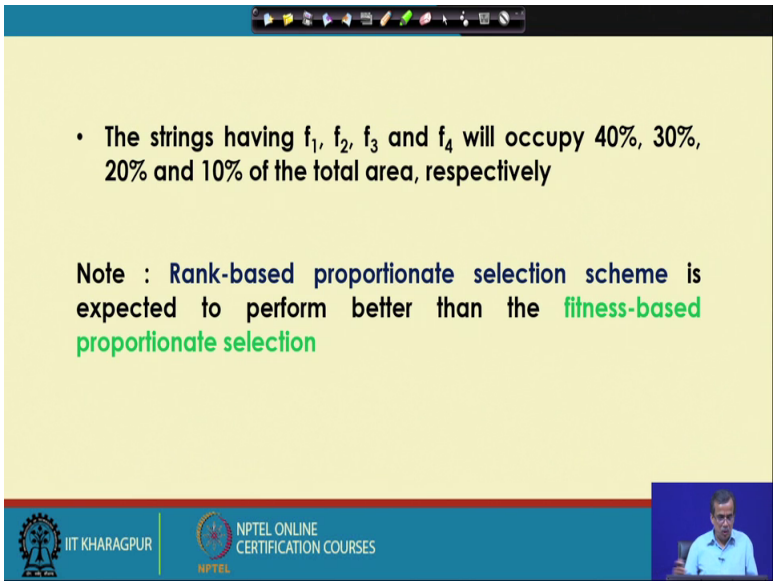
Now this ranking selection is once again one proportionate selection, but in terms of the ranks, but not in terms of only the fitness values. So, this is the difference between the ranking selection and the proportionate selection and generally we prefer the ranking selection because it has got less chance of premature convergence.

(Refer Slide Time: 17:28)



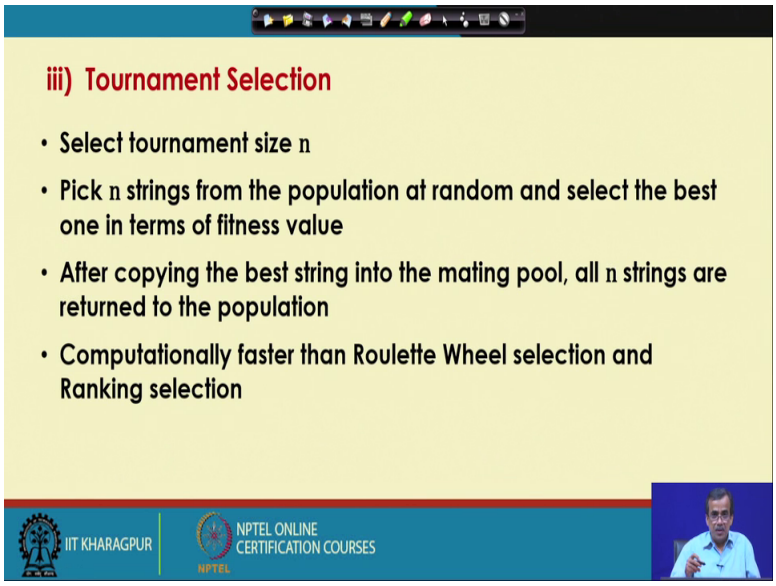
So, this is what I discussed.

(Refer Slide Time: 17:36)



Now, here I put one note that the rank based proportionate selection scheme is expected to perform better than the fitness based proportionate selection the reason I am already discussed.

(Refer Slide Time: 17:52)



Now, I am just going to discuss another reproduction scheme which is popularly known as the tournament selection and it has become very popular. Now a days we generally use the tournament selections only, now what I do is we have got the population size denoted by capital n we select the tournament size small n, now if capital n that is the population size if that is equal to 100 we take this tournament size small n might be equal to 3 4 5 something like this. Now once you have taken this tournament size. So, what I do is we take small number of solutions from the population at random using the random number generator and compare their fitness values.

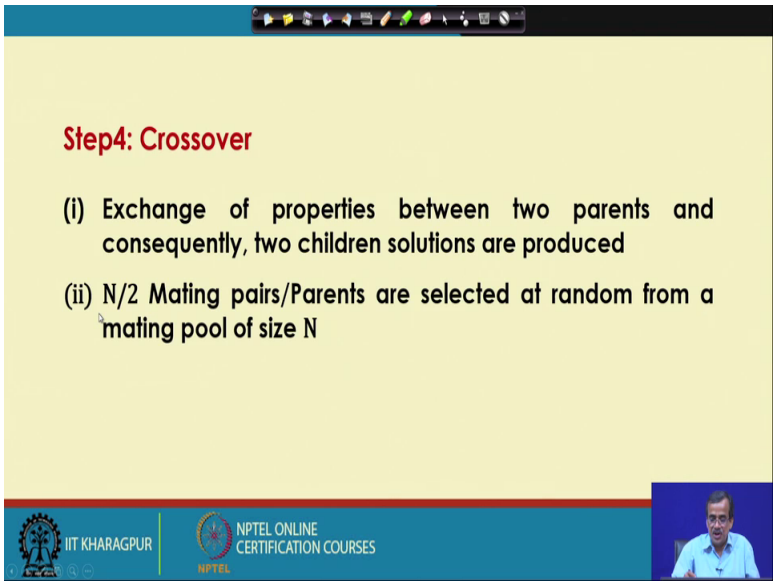
Now, supposing that I am solving 1 maximization problem. So, out of that small n of solutions the solution which is having the high highest fitness that will be selected in the mating pool and the all the solutions will be returned to the population of solutions. Supposing that the tournament size is a 3 out of 3 I have selected say the second 1 as the best. So, that I will copy in the mating pool, but all 3 solutions should be returned to the initial population and this particular tournament, I will have to play how many times capital n times and each time I am just going to select a particular the solution .

So, this is what I discussed; now this particular tournament selection is computationally faster than the roulette wheel selection and the ranking selection. Now another thing I just want to point out that GA on principle can solve the maximization problem. So, if you have the minimization problem as I discussed we will have to converted to the

maximization problem, but if we used tournament selection we did not convert the minimization problem to maximization problem and directly we can tackle the minimization problem using the principle of tournament selection.

Now, in tournament selection if you want to solve the minimization problem so out of small n number of solution you select that one which is having the minimum fitness or the lowest fitness in the mating pool. So, directly I can solve the minimization problem without converting it into the maximization problem, if I use the principle of this particular the tournament selection. So, this is one merit or advantage of tournament selection over the other selection skills.

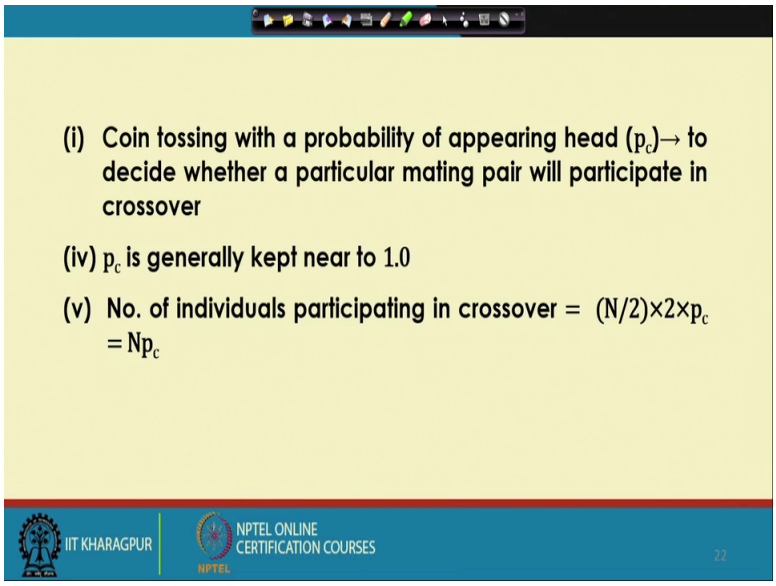
(Refer Slide Time: 21:20)



Now, then comes the concept of the crossover that is step 4. So, in crossover actually what we do is we exchange properties between 2 parents and consequently 2 children solutions will be created and some new properties will be coming due to this particular the crossover operator. Now how to implement this particular crossover operator to implement this particular operator what I do is we first try to select the mating pairs and each mating pair consists of 2 solutions.

So, out of the population size of capital n there will be N by 2 mating pairs at this N by 2 mating pairs will be selected at random. Now once we have selected this particular mating pair now I can go for the crossover operator and using this crossover operator I can find out the children solutions.

(Refer Slide Time: 22:32)



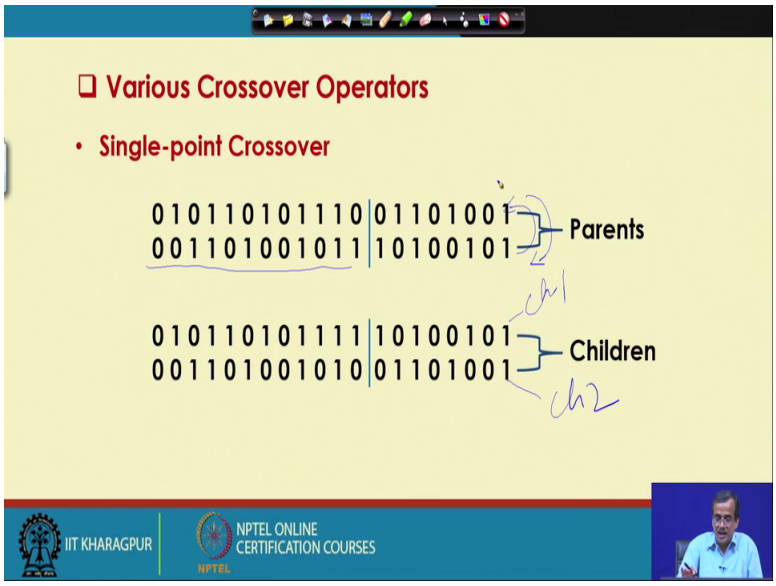
Now, what I do is how to implement this particular the crossover probability like how to decide whether a particular mating pair is going to participate in crossover or not.

Now, supposing that the probability of crossover is a 0.8, now if it is 0.8 to implement what we do is we try to take the help of some sort of coin tossing, now how to implement the coin tossing in computer program we implement with the help of 1 random number generator which will generate a number lying between 0 and 1, now if the generated number is found to be less than equals to 0.8. So, that is a success; that means, the head has come for the coin tossing.

So, what we do is we decide that that particular mating pair is going to participate in the crossover now generally this crossover probability we take in the higher range like starting from say 0.6 up to 1.0. Now let us see actually how to implement. So, this particular the crossover operator, now here the number of individuals participating in crossover can be calculated now supposing that the population size is capital N. So, there will be N by 2 mating pairs for each mating pair there will be 2 solutions. So, it is multiplied by 2 multiplied by p c. So, this is nothing N p c.

So, this N pc is nothing, but the number of individuals participating in crossover.

(Refer Slide Time: 24:32)



Now let us see how to implement the various crossover operators let me start with the single point crossover. Now here the principle is very simple supposing that this is the a particular mating pair and this is parent 1 this is your parent 2 and supposing that there are 20 bits, now if there are 20 bits there are 19 places for the crossover site selection and this particular crossover site is selected at random.

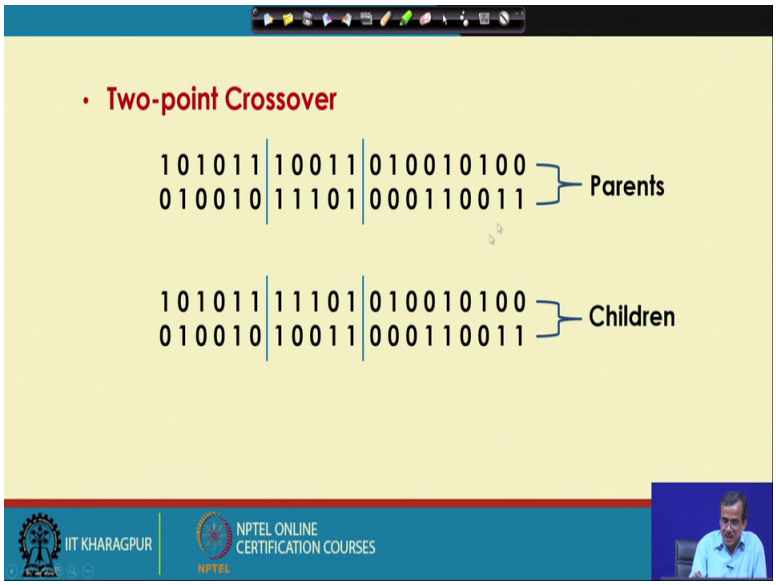
Supposing that this is the site which has been selected for the cross over at random, now if that is the crossover site according to the single point crossover actually what you do is the bits which a lying on the left hand side of the crossover site are kept on altered and whatever is there on the right hand side. So, we do the shopping now if I do the shopping what will happen is your. So, whatever is there at the bottom that will go to the top and whatever is there at the top? So, that will come to the bottom, but the left hand side of this particular the crossover site that will remain the same.

So, there will be no change now if you follow this particular principle then from the 2 Parents p r 1 and p r 2 I will be getting 2 Children. So, this is your child 1 this is your child 1 and this will be your child 2, now here I just want to mention that if I just keep the right hand side of the crossover site intact and if I do the shopping on the left hand side, I will be getting the same children solution only thing might be child 1 will be child 2 and so on that type of thing, but I will be getting the same set up 2 children, but on principle we do not disturb the left hand side of this particular crossover site.

The reason is the place value for the leftmost bit is much higher compared to the place value of this particular right hand side and on principle, we follow in GA the gradual search and that is why philosophically we do not consider the high change, but by following these 2 method anyone.

So, I will be by following these 2 methods you will be getting the same solution, but traditionally what we follow is whatever bits are lying on the left hand side are kept constant and whatever is lying on the right hand side we do the swapping. So, this is the principle which we follow for the single point crossover just to get 2 children solutions from the 2 parents. So, this is actually the second point the crossover.

(Refer Slide Time: 27:57)



Now, then comes the concept of the 2 point crossover, now here actually what we do is we select to crossover sites like your the crossover sites 1 and crossover site 2 once again at random and the principle is very simple. So, whatever bits are lying on the left hand side of the first crossover site and the bits which are lying on the right hand side of the second crossover site those are kept un altered and the bits which are lying in between the 2 cross sites. So, there will be swapping and if I follow this particular principle.

So, I will be getting if I follow this particular the principle. So, I will be getting the children solutions like this. So, here these are kept constant and all such bits are kept constant only thing whatever is lying in between the 2 crossover sites. So, there will be

swapping so starting from the 2 parents I will be getting 2 children solutions using the principle of the 2 point crossover.

Thank you.