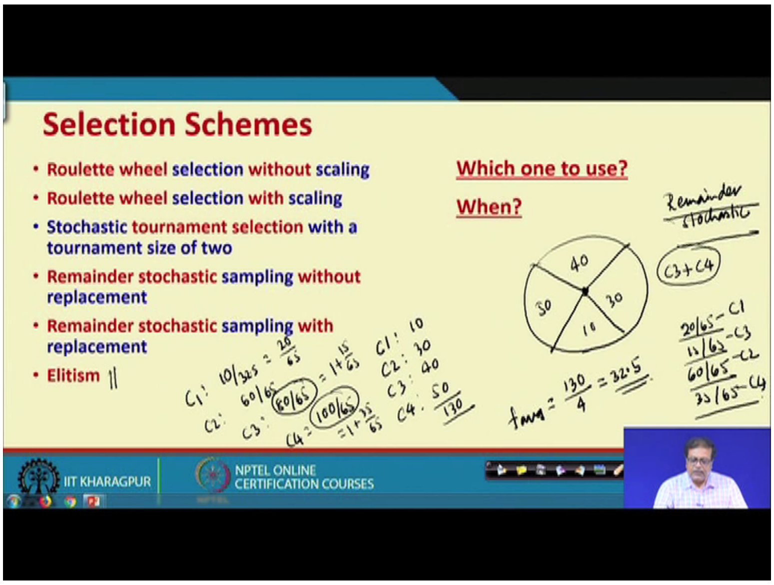
Selected Topics in Decision Modeling
Prof. Biswajit Mahanty
Department of Industrial and Systems Engineering
Indian Institute of Technology, Kharagpur

Lecture – 34
Genetic Algorithm Process (Contd.)

Right. So, in our course Selected Topics in Decision Modeling, we are now in lecture – 34, that is, Genetic Algorithm Process, we were discussing that and we continue in this lecture also. So, genetic algorithm process continued that is our 34th lecture.

(Refer Slide Time: 00:36)



Now, we were discussing about the different selection schemes and specifically I have discussed about the roulette wheel selection with and without scaling. See what happens, supposing I have certain chromosomes and those chromosomes has got certain fitness values then based on those fitness values, we create a roulette wheel and sometime to have a selection the balance between selection pressure and population diversity, we may have to scale the fitness values so that the population diversity improves. So, that we discussed in our last you know example, last lecture.

So, let us see what is this remainder stochastic selection process what happens just assume that I have four chromosomes only four chromosomes that is the population size and fitness values are 10, 30, 40 and 50. So, that total is 130. Now, we create a roulette wheel in such a manner. So, that you know this is the center. So, one will be 50, another will be 40, so, this is 50, this is 40, 30 and 10. Now, this is our roulette wheel when we have roulette wheel selection with or without scaling, first of all that is without scaling.

Now, with scaling these values are slightly modified to improve what is known as population diversity. See in remainder stochastic method what happens we take a slightly different method, what is that? So, we first take an average, see what is f average? f average will be 130 is the total and there are four chromosomes for strings sometimes they are called strings; So, 130 by 4. So, that is 65 by 2. So, 32.5, is it alright? So, you see now in C 1 fitness is 10. So, if I divide by 32.5 then this will be 20 by 65 C 2 will be 60 by 65, but C 3 will be 80 by 65 and C 4 will be 100 by 65.

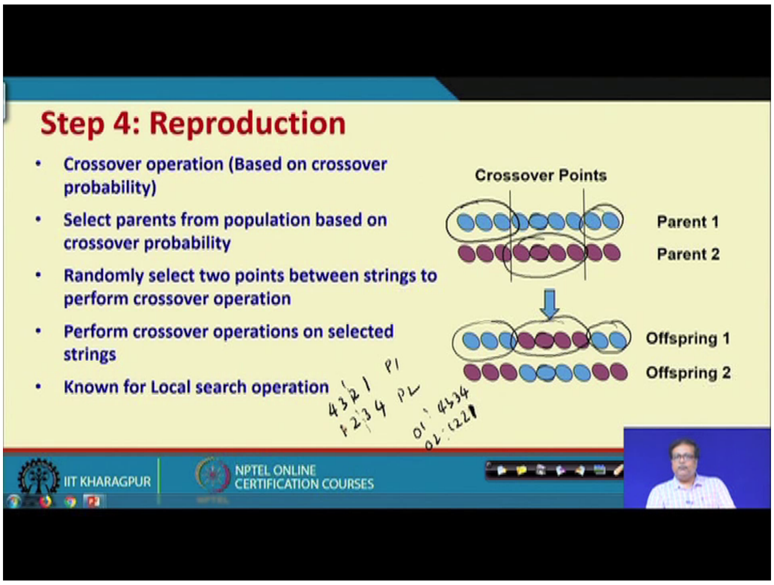
Now, in this case if you see C 3 and C 4 they are above average. So, you can actually call them as 1 plus 15 by 65 and this one 1 plus 35 by 65. So, you see C 3 and C 4 they have more than you know if I make f by f average then I get you know one string plus some more. So, what we do in remainder stochastic remainder stochastic method selection you know we take one string of C 3 and one string of C 4, right, one each because there is 1. And then I create the roulette wheel in such a manner that which is 20 by 65, 15 by 65, so, the roulette wheel will be in this ratios 20 by 65, 15 by 65 then your 60 by 65 and 35 by 65. So, this was C 1, this one is C 3, this is C 2 and this is C 4.

So, in that ratio I create the roulette wheel and then I have already got one C 3 and one C 4 and then through the roulette wheel may be I take two more strings. So, I get four strings for selection purpose, is it alright? Although we may not like to take four strings we may take only two strings. Now, please remember sometimes you know it is not really a necessary that we just follow the roulette wheel or some specific method.

We might like to make use of what is known as the tournament selection. So, you see that out of these roulette wheel I select two, but then again I make a tournament I mean out of the two we see the higher fitness and we take that. So, that is called the tournament selection is it alright. So, I just take two strings and then have a tournament between the two, the better one we selection, that is a tournament selection.

The elitism is specific type of selection where a selected chromosome is directly moved to the next generation, right. So, there is no cross over or mutation. So, that is what is the essentially the selection method for the different selection method that we have.

(Refer Slide Time: 06:39)



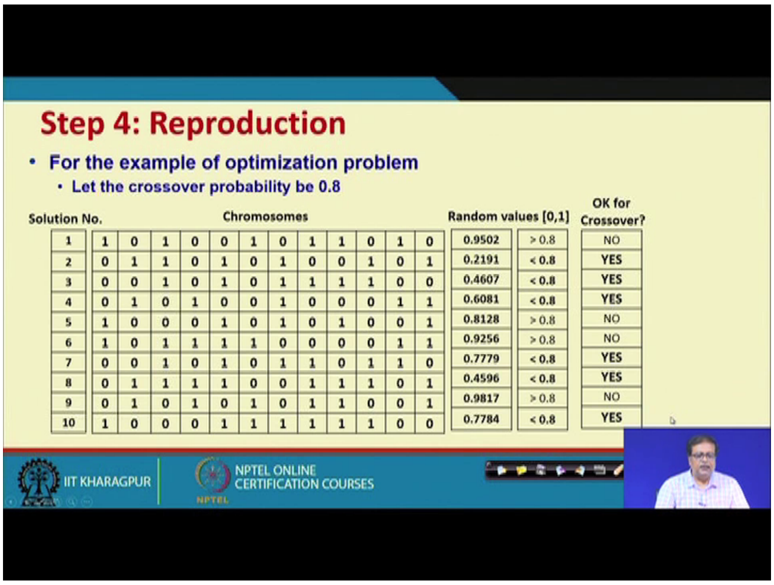
Now, after this let us see, what are the various reproduction possibilities. So, there could be crossover operation based on crossover probability, select parents from population based on crossover probability, is it alright? Randomly select two points between strings and perform crossover operations then perform crossover operations on selected strings and known for local search operation. So, essentially the crossover operation is something like this. So, you see there is a crossover point.

So, just look what is happening. There is a parent 1, there is a parent 2, right. So, I first of all I have to have a crossover probability. Now, based on the crossover probability there will be there could be crossover or there could be no crossover, right, supposing we select a parent where the crossover probability is very low you know beyond a certain crossover suppose the crossover probability is 0.7 and when you select a particular parent where the probability is below 0.7 then there will be no crossover, alright.

So, you know so, I mean the 70 percent time we shall do crossover, 30 percent time we may not do crossover. So, that is the 70 percent means. So, supposing the probability is you know not within that 70 percent then we do no crossover. But, when you do crossover we can choose one or more sites. So, here we have taken two crossover sites so, which is again randomly decided. So, this is crossover site 1 and this is crossover site 2. So, you see how the crossover is made the blue ones the first offspring the first three you know it inherent from parent 1. So, this is these are the these three, then two also it takes from the parent 1, but from parent 2 it takes these four. So, you see these four are coming from parent 2. So, that these becomes offspring 1 and these becomes offspring 2, the remaining ones will constitute the offspring 2.

So, this is an example of the crossover. Now, it is not necessary that all crossover should be like this right you can also think of a crossover supposing I have two strings one is 4 2 3 1 another is 1 2 3 4. So, I can have just a single crossover site in the middle. So, this is parent 1 and this is parent 2 then what will be my you know two offspring's O 1 and O 2 simple it should be 4 3 3 4 and 1 2 2 1, is it alright so, 1 2 2 1. Now, these are definitely not permutation encoding, this is just for any numbers. So, that is how you can have different crossover sites and different kinds of crossover.

(Refer Slide Time: 10:08)

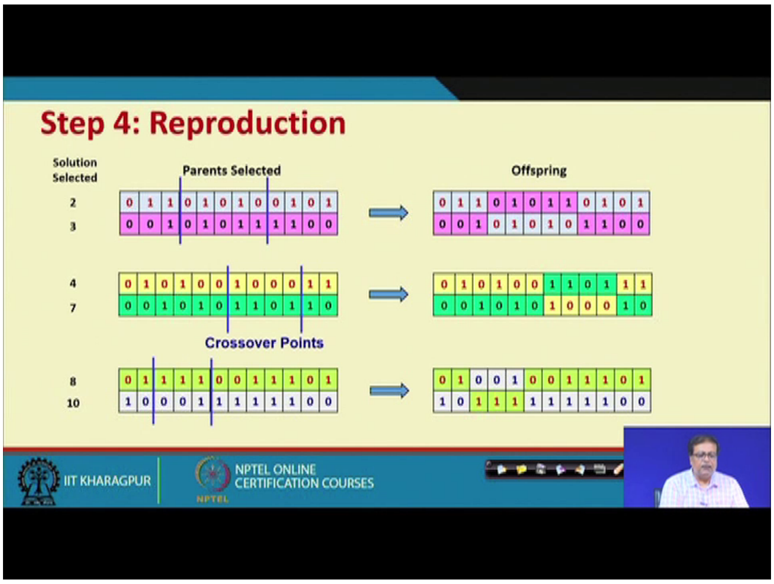


Now, apart from that we can also have the mutation, but before that let us look at these example. So, you see I had those 10 strings. So, these are my 10 strings and those 10 strings the first chromosome random value is 0.9502, then you know these 0.2191 what are these? These are some random numbers generated between 0 and 1, is it alright?

That assume the crossover probability is 0.8. So, so what you can see that out of these chromosomes the probability which is below 0.8. So, if it is up to 0.8 we do crossover above 0.8 we do not do crossover. So, this is how we define. So, those which are below 0.8 so, you know 1 2 3 4 5 and 6. So, you see six of them have come yes, they are for crossover and remaining ones they are not for crossover, is it alright. So, that means, those chromosomes if they are selected there will be crossover and otherwise there will be no crossover, is it alright.

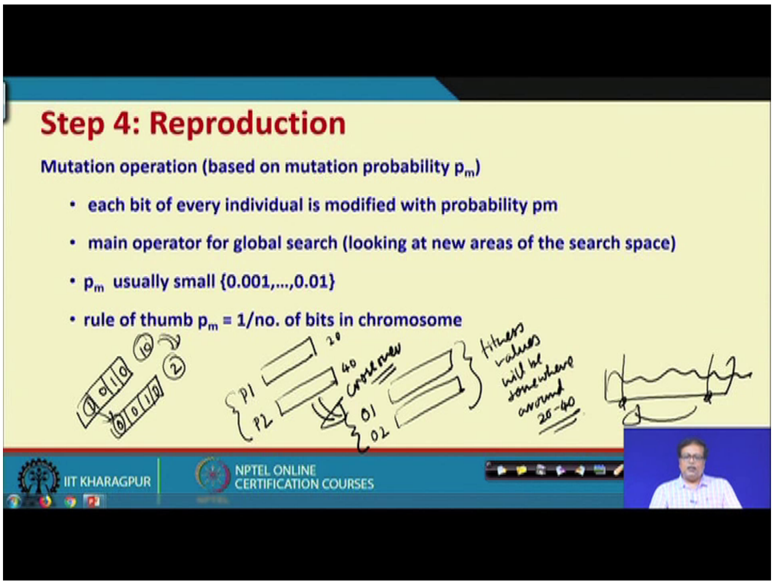
So, that is how the based on the crossover probability we can choose that which chromosomes will be selected for crossover and which chromosomes will not be selected for crossover. So, you can understand what we these are my original 10 chromosomes in the initial population we have taken 10 random numbers. And based on the random numbers we have checked against the crossover probability and selected the strings either for crossover or not for crossover. So, that is the reproduction.

(Refer Slide Time: 12:04)



So, here is an example. So, supposing in these 2 and 3 and this is how the offsprings are created; so, 0 1 1 then 0 1 0 1 1 then 0 1 0 1 here 0 0 1 then 0 1 0 1 1 and 1 1 0 0. So, these two again they crossover sides are different. So, you can see the offsprings and 8 and 10 again you see these are the crossover sites. So, they are actually crossed over. So, this is how the offspring’s are obtained from the parents, is it alright. So, these are some examples of crossover.

(Refer Slide Time: 12:48)



Now, the mutation on the other hand is something which is essentially used see like there is a basic difference between the crossover and the mutation. What crossover essentially does, suppose we have a parent and 2 parents so, if the parent has a particular fitness and parent 2 has a particular fitness so, the offspring's for the child's which you get from these two parents, they will be somewhat like them.

Now, what is exactly meaning of somewhat like them suppose the fitness value of P 1 is something like 20 fitness value of PP P 2 is something like 40. Then usually seen that the fitness values for the offspring will be somewhere around 20 to 40 somewhere around not exactly depends on problem to problem, obviously, we cannot generalize like this.

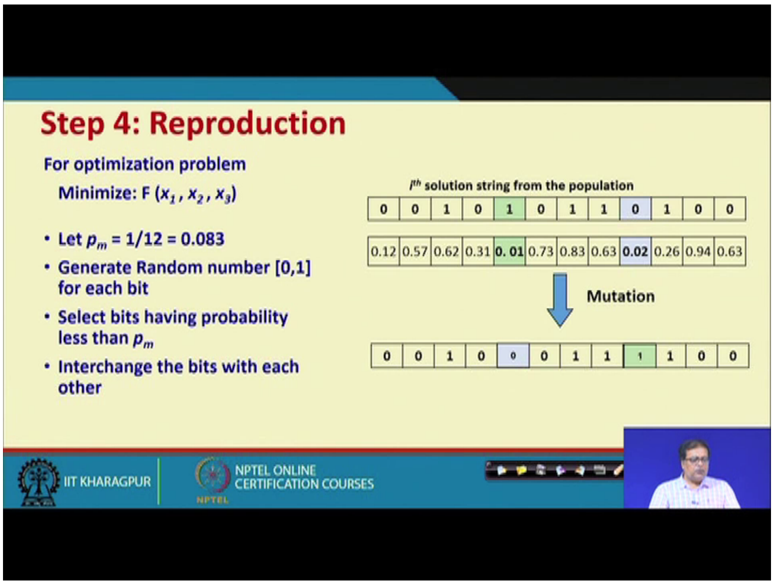
But, then in a general sense we can say that if the offspring's are created by crossover then the fitness of the offspring and the fitness of the parents there will be somewhere near, in a sense the offspring will be very much like parents right not exactly same, but definitely different, but somewhere around them. On the other hand what mutation does mutation completely changes, completely changes. Say for example, let us say you know you have a number supposing just say 4 1 0 1 1 alright, say it evaluates to 8 plus 2, 10. Supposing you make the first one as 0 then it becomes 0 0 1 0, it becomes 2, just imagine.

So, by mutation I change this 1 to 0, now as I change these 0 to 1 or 1 to 0 the value comes from 10 to 2. So, it is a drastic change. So, in other words in the solution see this is my solution space, the number which was somewhere here now it has come somewhere here. So, you know it completely moves to another portion of the search space and if the search was getting localized you know it increases what is known as population diversity.

So, that is how mutation is used the mutation is used to really give a boost to population diversity where the solution might be getting localized, right. So, that is the essential advantage of mutation as an operator is it alright. So, the mutation operator, but then usually the probability of mutation is kept very low, we do not want mutation all the time. We want mutation some of the time and so, that we can increase population diversity particular when the solutions are getting localized, but keeping in mind that mutation you know too much mutation will again throw the selection pressure out of gear, where the salutation was converging the convergence may be lost

So, that is the disadvantage of a very high value of mutation, right. So, mutation probability should be low in fact, very low. In fact, you see it is somewhere between 0.001 to 0.01, is it alright. So, each bit of every individual is modified with probability p m. So, there are so many bits. So, you can modify anyone of the bits with a very low probability, is it alright. So, sometimes we used 1 by number of bits in chromosome, but then usually the probability of mutation should be very low indeed, right. So, having known that let us take an example of the mutation.

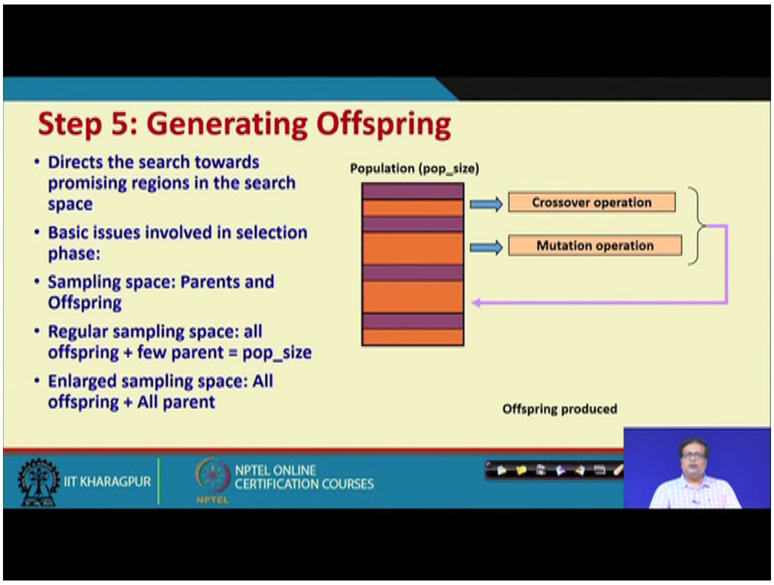
(Refer Slide Time: 17:40)



So, here you see that supposing we have optimization the minimize this function now i-th solution strings from the population. So, you can see that the let the pm becomes 1 by 12 that is 0.083. So, generate random number 0 1 for each bit. So, you know for each bit we have created random numbers. So, 0.12, 0.57, 0.62, 0.31, 0.01, so, in this case you can see these two places the probability is lower than this, right. So, what happens we can select bits having probability less than p m. So, we have selected two bits and then interchange the bit. So, you see one such mutation could be interchange of bits or you can also do 1 to 0 or 0 to 1.

So, as I said there are different designs that are available, there is no standard design all the time there are different designs and different designs are applicable in different places, is it alright. So, in this particular case we have gone for interchanging of bits. So, you see the interchange is happening, right. So, what has happened the 1 has become 0 and 0 has become 1, is it alright. So, that is the kind of mutation that we have introduced in this particular case, right.

(Refer Slide Time: 19:22)

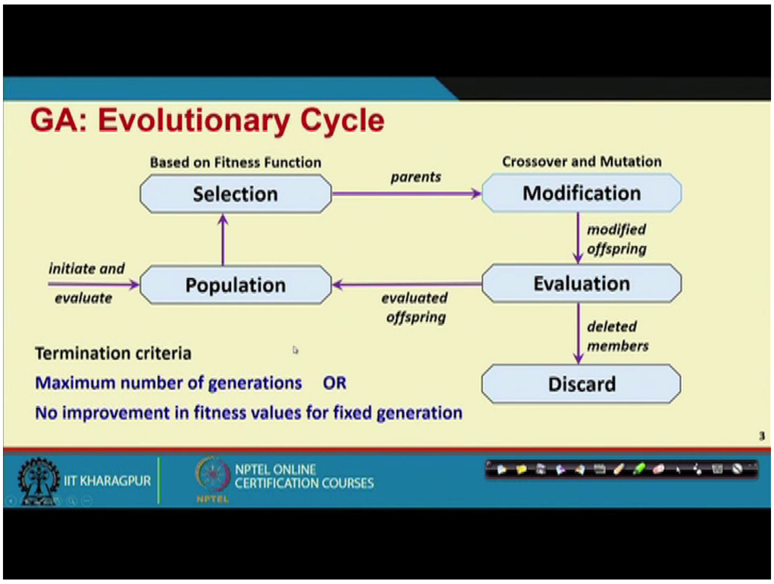


So, having seen that now let us see how the offspring is generated through crossover and mutation. So, you can see just now you have seen the animation we will explain and we will see it again. So, you see what really happens they generate offspring's, they direct the search towards promising regions in the search space basic issues involved in the selection phase.

So, you see what really happens is that we have a population, right, so, that population through crossover and mutation you know some of the population they changes, the remaining population we can keep may be as it is. Suppose, we assume to do like this that either we create the new population only through offspring's or we select some of the population remaining population we keep unchanged and that population through a crossover and mutation you know we make a change. So, just see this animation once again. So, we have a population now there is crossover and mutation through a crossover and mutation we create the offspring and that offspring replaces those parents, is it alright.

The remaining population remains as they are, so, we have the new generation, is it alright. So, that is how we generate the offspring's, is it alright. So, that is how exactly what happens. So, let us see these the very beginning we have this in our previous lecture.

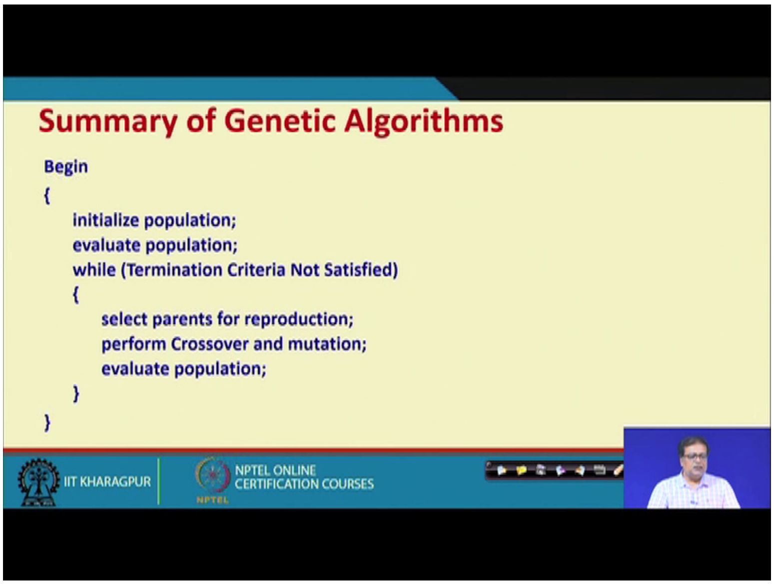
(Refer Slide Time: 21:08)



So, we have the initial population then evaluated offspring I mean selection, modification, modified offspring and then evaluated offspring joins the population, right. So, essentially precisely that is what we have seen once again in these particular case, right. So, that is how we generate the offspring once again, is it alright. So, probably you understood the process that how we generate from a given generation population to the next generation population through the process of selection and recombination in the sense of crossover and mutation.

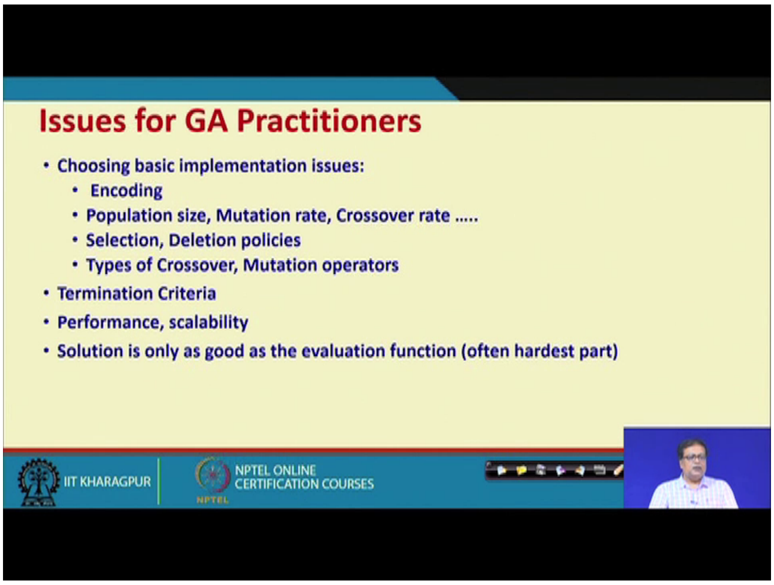
Please remember, while we do crossover and mutation we always keep in mind the what is known as the probability of crossover and probability of mutation that we always keep in mind, right.

(Refer Slide Time: 22:13)



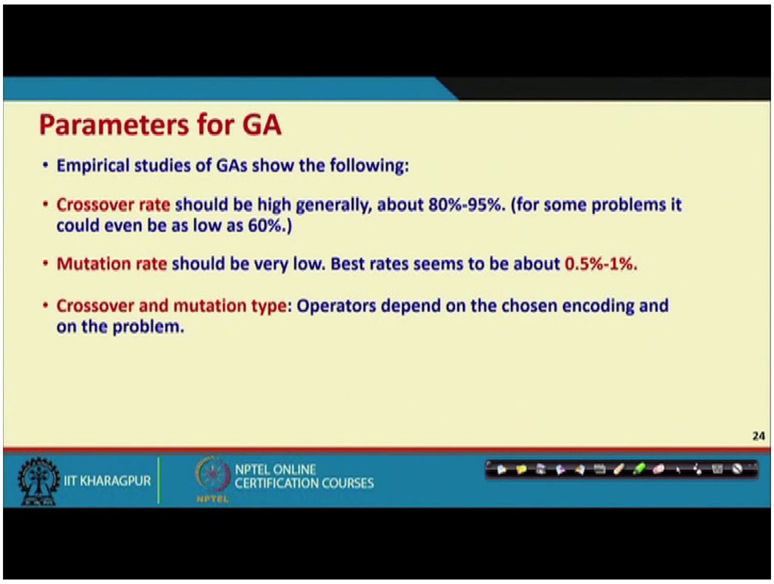
So, this is the summary of genetic algorithm; initialize population, evaluate population, while termination criteria not satisfied, select parent for reproduction, perform crossover and mutation, then evaluate population, right. So, this is how the genetic algorithm process goes on, right.

(Refer Slide Time: 22:38)



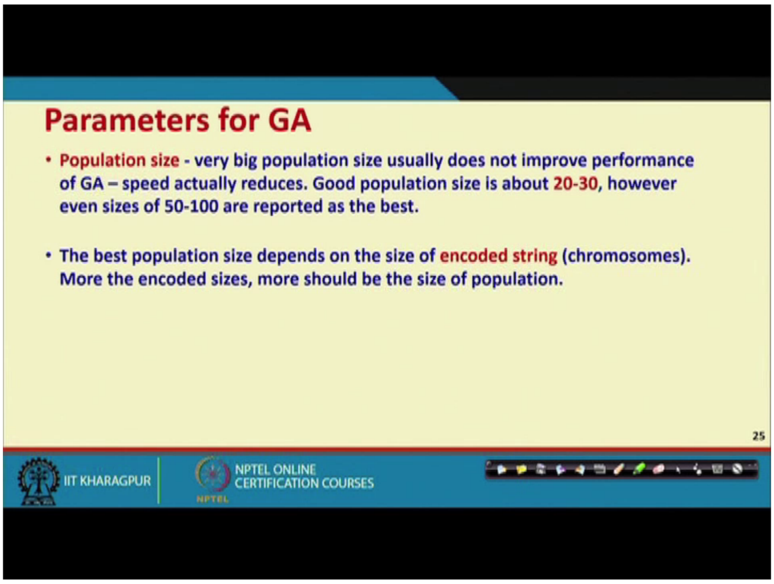
So, issues the issues of encoding, population size, mutation, crossover rate, selection, deletion policies, type of crossover, type of mutation, termination criteria, performance and scalability means for a bigger problem can this method be used and salutation is only as good as the evaluation function, is it alright. So, every genetic algorithm is a different problem and should be seen accordingly.

(Refer Slide Time: 23:13)



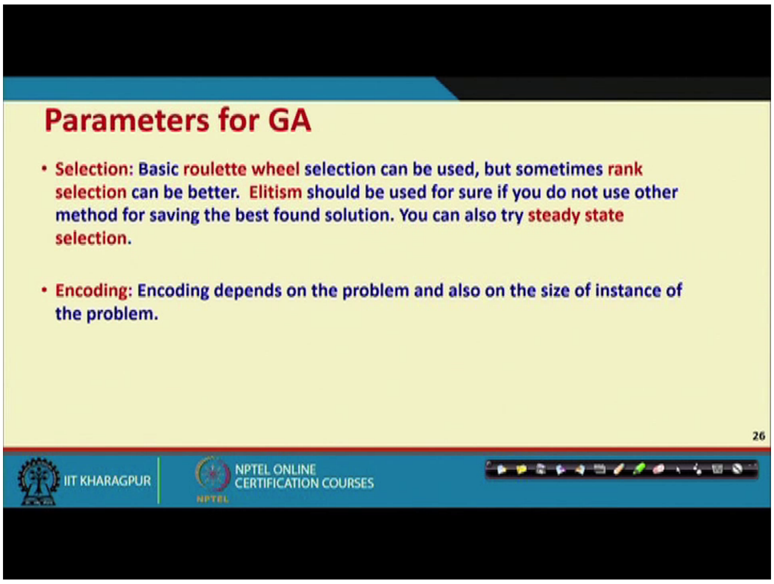
There are different parameters that are used like crossover rate which should be on the high side, the mutation rate should be low and crossover and mutation type again depend on the encoding and the kind of problem that we have taken.

(Refer Slide Time: 23:29)



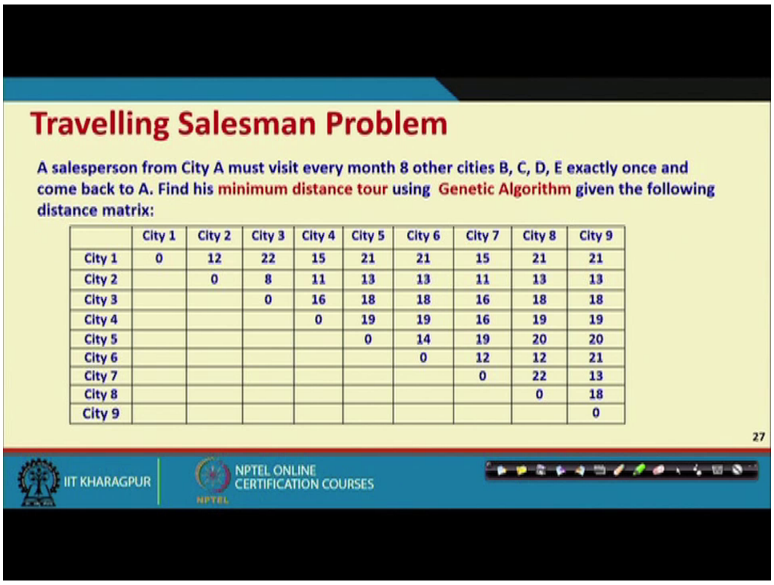
The very big population size is not good, speed reduces the good population size is 20 to 30, but for big problems we may even have 50 to 100. In fact, size of encoded string you know more or less decides the population size. So, more is the encoded size, more should be the size of the population.

(Refer Slide Time: 23:54)



So, selection sometimes basic roulette wheel can be selected, sometimes we can even use rank selection means the 1 2 3 4 5 replace rank fitness by rank and do the roulette wheel, then elitism should be used; elitism means some population the good ones directly transferred to the next generation, right. There could be steady state selection also the selection done in a steady state manner, then encoding depends on the problem and also on the size of the instance of the problem.

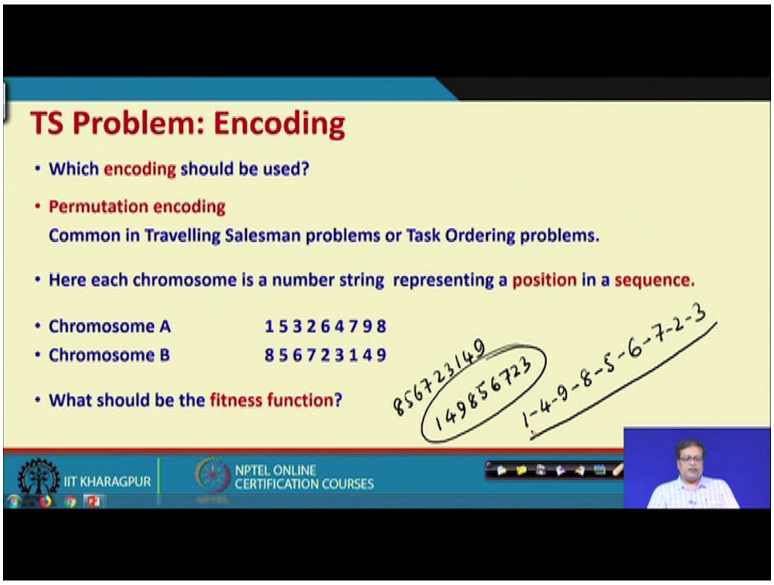
(Refer Slide Time: 24:30)



Let us see an example very quickly on the travelling salesman problem. So, let us say we have a problem of you know a travelling salesman problem where we have 9 cities, right. There are 9 cities and between 9 cities we have a sales person who moves from city 1 to all the other cities exactly once and comes back to the city of origin, that is city 1. Now, here is the distance matrix between city 1 to city 2 and all the way up to city 9. So, it is assumed that say city 1 to city 22 means city 3 to city 1 is also will be 22, is it alright.

So, what we have to do find the minimum distance tour using genetic algorithm given the following distance matrix. So, how do we go about it? So, the first task is encoding. Now, look here what exactly we are interested is in a tour. So, basically in a tour you know every city should be exactly once. So, 1 to 2 to 3 to 4 to 5 to 6 to 7 to 8 and again 8 to 1, but, that is imperative there is no need to write it.

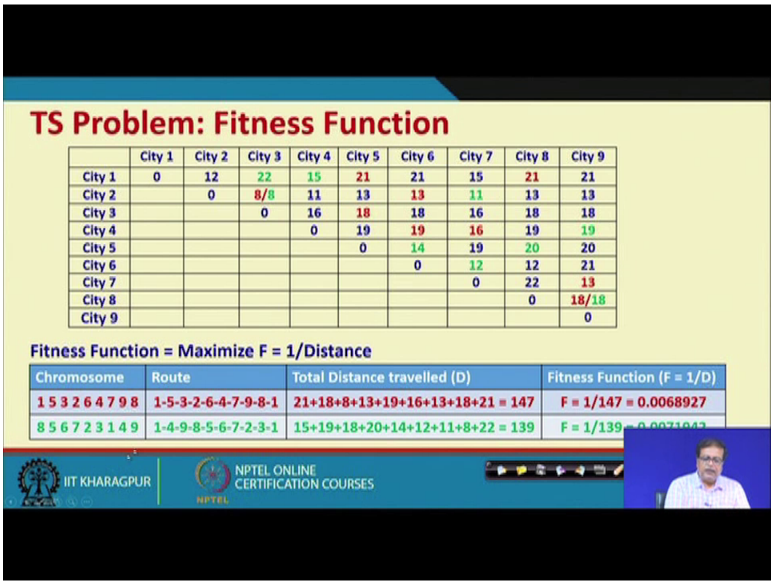
(Refer Slide Time: 26:02)



So, how do we do encoding? So, as I have told before the correct encoding here is can be called as permutation encoding common in travelling salesman problem or task ordering problems. So, each chromosome is a number string representing a position in a sequence. So, look here two chromosomes 1 5 3 2 6 4 7 9 8 and 8 5 6 7 2 3 1 4 9 they are valid representations because you know they are each string is exactly once.

The fact is which is interesting, please note here that this 8 5 6 7 2 3 1 4 9 is actually same as 1 4 9 8 5 6 7 2 3. Basically, it means 1 to 4, 4 to 9, 9 to 8, 8 to 5, 5 to 6, 6 to 7, 7 to 2 and 2 to 3, that is what it means. So, what should be the fitness function? The fitness function in this case can be evaluated in these manner.

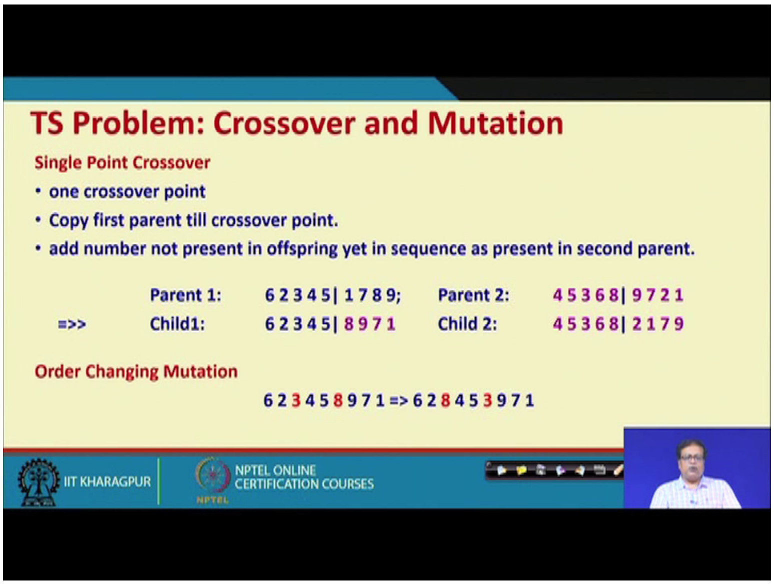
(Refer Slide Time: 27:29)



So, what is we can find out the total distance travelled and fitness function can be taken as one by the total distance travelled. So, usually you see most of the time we try to solve the problem as maximization problem rather than minimization problem, right that is usually done. So, how to convert these problem to a maximization problem? So, minimize distance. So, maximize 1 by distance that we can define as a fitness function.

So, you see this is the chromosome this is first chromosome this is the root, this is the second chromosome this is the root. So, from the matrix we found out the total distance travelled in this case 147, in this case 139 and 1 by that will give us the fitness function value, right. So, we can find out the fitness function in these manner.

(Refer Slide Time: 28:29)



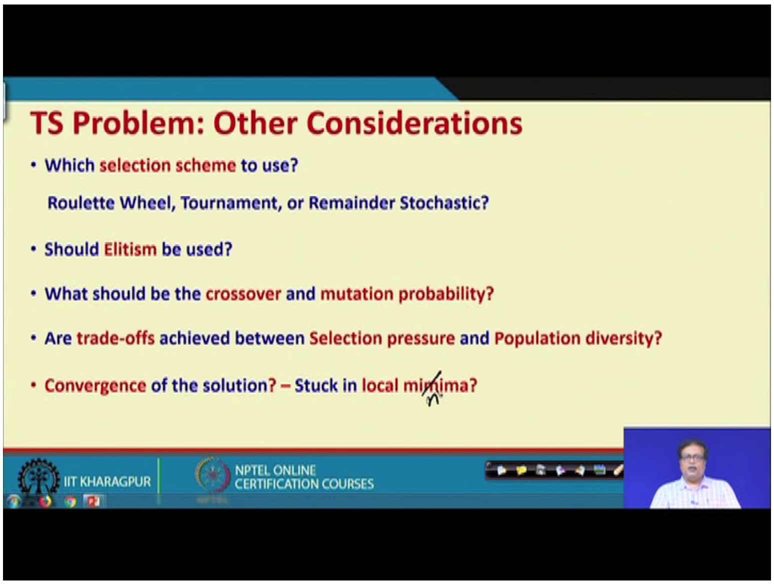
Now, what kind of crossover we are going to use here. So, you see we should be very careful because you know we should have one number exactly once. So, we just cannot do the usual crossover and mutation as the kind of examples we have seen in the previous problem. See, suppose, this is my parent 1 and this is my parent 2. So, you see we decide a crossover point.

So, supposing crossover point comes here. So, copy first parent till crossover point. So, child 1, so, look here. So, this is parent 1. So, this portion just copy and then number not present in offspring yet in sequence as present in second parent. So, what are the numbers not present? 1 7 8 9; Look is look their position in the second parent. So, this is 1, this is 7, this is 8 and this is 9. So, 8 9 7 1 in that order; So, you see 8 9 7 1, that is child 1, right.

Similarly, for child 2 look the method that 4 5 3 6 8 directly copied. What are not having? 9 7 2 1. So, C 9 7 2 1 in second parent; so, this is 9, this is 7, this is 2 and this is 1. So, in that order 2 1 7 9 2 1 7 9, is it alright. So, this is the kind of crossover which is suitable for such kind of permutation encoding kind of problems. And, order changing mutation so, you see what is we arbitrarily see two numbers where the mutation probability is low enough.

So, supposing these number and this number. So, simply interchange those numbers. So, 3 becomes 8, 8 becomes 3. So, that in the resulting chromosome each number has appeared exactly once, is it alright. So, understood the crossover and mutation? Once it is understood next is some of the other considerations.

(Refer Slide Time: 30:58)



So, which selection scheme should we use? roulette wheel, tournament, remainder stochastic, right. We can choose between them depending on the problem. Should elitism be used? We can also use elitism. Some of the good population could be directly transferred to the next generation. So, crossover probability as you see should be high enough and mutation probability should be low.

There should be tradeoff between selection pressure and population diversity, is it alright and then finally, convergent of the solution should be seen also and we should not see that stuck in local minima should local minima should not be there, right. The solution should not get stuck in local minima, is it alright.

So, those are the issues which should be kept in mind and based on which we have to convert actually convert these to a computer program and through a programming method we must try to find the solution, right. So, we stop here and you know in next class we shall see more examples of genetic algorithm, right.

So, thank you very much.