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

Lecture - 31
A Practical Optimization Problem (Contd.)

So, we have got the fitness information for the whole population of this particular the ga, and now enough to concentrate on the modified fitness values, now if I concentrate on the modified fitness values, now the modified fitness value for the third and the 4th, they are very high, but our aim is to minimize this particular the objective function. So, these 2 solutions will not be a good solution and we want to delete these 2 solution, in the mating pool.

(Refer Slide Time: 00:22)

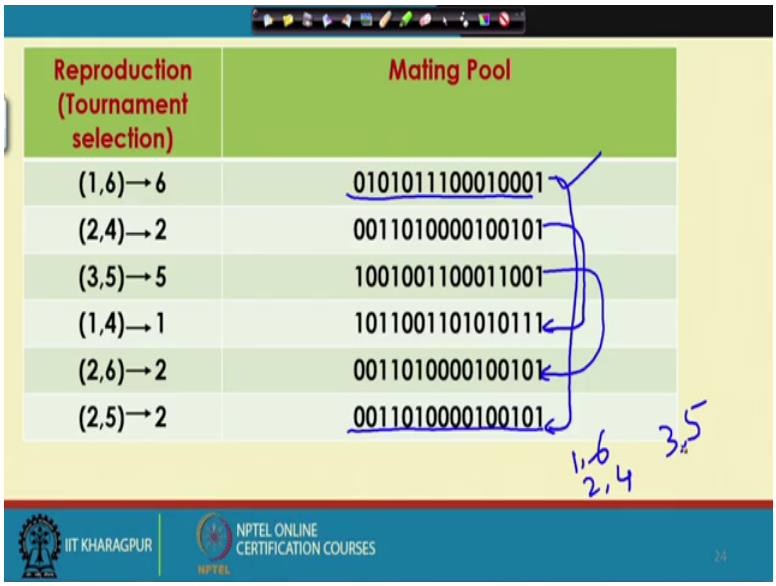
Real Value for b	Real Value for h	Fitness f	Penalty P	Modified Fitness (F)
0.142	0.069	15.40	0.0	15.40
0.045	0.012	0.85*	0.0	0.85
0.019	0.007	0.22	10.58×10^{24}	10.58×10^{24}
0.019	0.008	0.23	5.35×10^{24}	5.35×10^{24}
0.117	0.057	10.48	0.0	10.48
0.071	0.056	6.25	0.0	6.25

* Minimum fitness of feasible solutions

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES

Now, to get the mating pool we take the help of reproduction scheme, now here we are just going to use, the concept of the tournament selection as a reproduction scheme. Now, to establish the principle of this particular, your the tournament selection I take the help of this modified fitness values, now let us see how to use that particular the concept of modical modified fitness values.

(Refer Slide Time: 01:29)



Now, here in this tournament selection actually what we do is, we try to compare the fitness values and before that, actually we will have to decide what should be the size of the tournament? There are 6 solution here, 6 ga strings here. So, I will have to play this particular tournament 6 times, now at each time I am just going to pick 2 solutions out of this particular 6 at random.

Now, in the first tournament supposing that I have selected, say the first solution that is the first ga string and the 6th ga string, and I am just comparing their fitness values. Now, if I compare the fitness values, the modified fitness values for the first and the 6th. So, I will be getting the minimum fitness is for the 6th one, 6th ga string. So, I am going to select the 6th ga string; that means, the 6th ga string from the initial population will be directly copied here.

Now, similarly I just go for the second tournament and once again the tournament size is equal to 2. So, I am just going to select the second and 4th at random and I am just going to compare their modified fitness values, and the modified fitness values for the second one is better, that is the smaller in numerical value. So, that is selected here. So, from the initial population, I am just going to select the second ga string in the mating pool. The third tournament, so, I am just going to select third and the fifth at random and I am just comparing their fitness. Fifth one is selected, because the numerical values is less

compared to that of third. So, fifth one is selected. So, the fifth ga string is copied in this particular the mating pool.

Next, I go for the 4th tournament, that is and I select first and 4th at random and out of this first and 4th, the first one is better. So, I am just going to directly copy it here, next is the fifth tournament, we select second and 6th at random and I am just going to select second, because it is modified fitness values is less compared to that of 6. So, I am just going to copy here, the second ga string in the mating pool.

Next, I concentrate on the last tournament, that is the 6th tournament and we are selecting the second and fifth, and out of this second and fifth the second one is better. So, I am just going to keep a copy in the mating pool that is the second ga string. Now here one very important thing are to be noted, that if you find out a very good string, there is a possibility there could be a multiple copy of that particular ga string, in the mating pool. For example, we can see that the second ga string has been copied thrice; the population size is only 6 and out of 6; see the certain ga string we have copied thrice. So, this is the second ga string, second ga string, second ga string.

Now, the reason is, if you see the fitness value, in terms of the fitness value if I see, now the second ga string is actually, the is a very good ga string in terms of the fitness value. If you see this is 0 point 8 5. In fact, the second ga string was the best, in the initial population and we have got 3 copies of this particular the ga string in the mating pool and once, we have got this particular the mating pool. Now, I can proceed with the next operator, but before that another thing I just want to tell like, if I just concentrate on the bad ga string, that is the third one and the 4th one, the third one and 4th one we can see that there is no copy, of the third one and the 4th one in this particular the mating pool.

Because, these 2 ga string are found to be worse and those have been deleted, through this particular reproduction scheme, from the mating pool and this mating pool is actually, is a better population of solution and if I see the average fitness of this particular, the population of this mating pool. The average fitness of the mating pool is expected to be better compared to that, of the initial population.

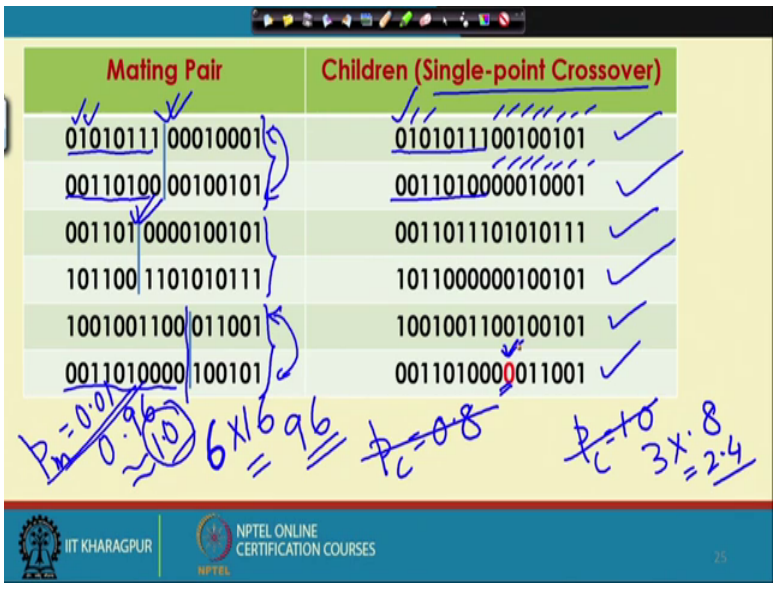
Now, once you have got this particular mating pool, now you will have to find out the mating pair, now how to find out the mating pair? Now, to determine the mating pair actually, what you will have to do is, in a one mating pair there will be only 2 ga string.

Now, how to determine the mating pair? Supposing that I am just going to decide, what should be the mating pair corresponding to the first ga string in the mating pool. So, this is the first ga string in the mating pool.

Now, this mating this particular ga string is going to meet with the other and that particular selection has to be at random, now what you can do is. So, I can take some random mating pair for example, say I am taking that one is going to meet with the 6th one. So, in the first mating pair, I am just going to take a copy of this and then, I am just going to take a copy of this. So, the first mating pair that is one is going to meet with 6, next I will have to concentrate on this particular the second one, now the second ga string can meet with third 4th 5th and let me consider, let me assume that the second one is going to meet with the 4th one, this selection is also at random.

So, second is going to meet with 4th and now, I will have to concentrate on the third one. So, the third one is going to meet with the remaining, that is the fifth one. So, third is going to meet with the fifth. So, these are going to form the mating pairs and as I told, this mating paired selection is at random, but here care should be taken that in a particular mating pair, both the ga string should not be exactly identical. Now, if you get that both the mating ga strings are identical, there will be no diversification in the properties in the children solution. So, that has to be avoided.

(Refer Slide Time: 09:01)



Now, once you have got this particular the mating pair. So, I can write down this particular mating pair. So, this is nothing but, the first mating pair and this is the second mating pair and the third mating pair. Now, I will have to decide whether each of these particular mating pairs, is going to participate in crossover or not, now here the probability of crossover that is taken to be equal to 1.0 and here we have got only 3 mating pairs and if this is the situation; that means, all the 3 mating pairs are going to participate in this particular the crossover; that means, all 3 mating pairs are going to participate.

Now, let me take a very hypothetical situation, supposing that the pc is not equal to 1.0. So, pc is equals to 0.8, now if pc is equal to 0.8 out of these 3 mating pairs. So, on an average 3 multiplied by 0.8, that is 2.4 mating pairs are going to participate in crossover; that means, either 2 or 3 probabilistically, mating pairs are going to participate in crossover, but here fortunately we have taken that pc is equal to 1.0. So, all 3 mating pairs are going to participate in crossover.

Now, how to implement in computer programming? It is very simple; we take the help of a coin tossing and this, if through this particular coin tossing. So, this coin tossing actually we will have to implement, with the help of a random number generator. Now this random number generator, is going to generate a number lying between 0 and 1, now if it can generates a number which is less than equals to 0.8. So, that is a success as if head has come, through this particular coin tossing and if it is a success that this particular mating pair is going to participate in crossover.

Now, this particular crossover probability is generally implemented in computer program, with the help of a random number generator, generating the number lying between 0 and 1. Now, supposing that all 3 mating pairs are going to participate in crossover and once they are going to participate, now I am just going for the single point crossover. So, for this single point crossover, I will have to find out the crossover side at random.

Now, here I have got 7 plus 9 there are 16 bits in a particular ga string. So, I have got 15 places for this particular the crossover side selection, and this particular crossover side selection. So, that is also done at random, using the random number generator. Now, supposing that for the first mating pair, the crossover side has been selected here and this

is a single point crossover and whatever bits are lying, on the left-hand side of this particular crossover side. So, those things will remain intact here like 0 1 0 1 0 1 1 1 0 1 0 1 0 1 1 1. So, up to these, this is kept intact similarly here, up to this this has been kept intact, there is no change. So, these bits are lying on the left-hand side, of this particular crossover side and the bits, which are lying on the right-hand side. So, there will be a swapping. So, this will go up that is 0 0 1 0 0 0 0 1 0 0 1 0 1 and this particular thing will come down 0 0 0; that means, 0 0 0 1 then 0 0 0 1. So, this will come down.

So, starting from these 2 parents of first mating pair, I will be getting 2 children solution and following the same principle, for the second mating pair. So, this is the single point, this is your the crossover side and using the signal point crossover. So, I will be getting these 2 children solution and for the third mating pair. So, this is the crossover side. So, the bits which are lying on the left-hand side of the crossover side will remain same and there will be swapping of the bits lying on the right-hand side, of the crossover side. So, it will be getting these and these at the 2 children solution.

So, this is actually how to implement the single point crossover, to get the children solution from this particular the mating pair. Now, I will just go for the next operator, that is called the mutation and here, actually I am just going to use the bitwise mutation. Now, before I go for the bitwise mutation, let me count the number of bits which we have on this particular the mating pool or the mating pair, now in 1 ga string, I have got 16 bits and similarly, I have got 6 our ga string. So, 6 multiplied by 16 that is nothing but your the 96. So, we have got 96 bits.

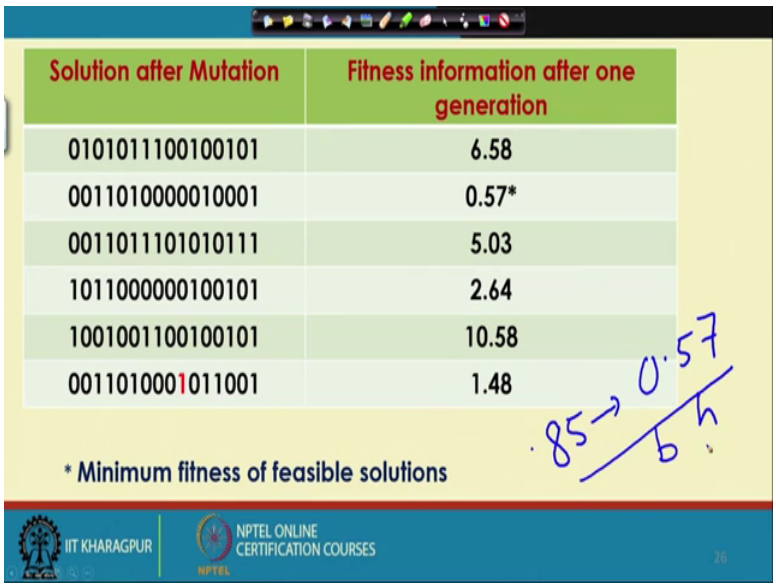
Now, if I take the probability of mutation that is pm is if I take 0.01. So, what I can do is? I can multiply 0.01 with this 96 and I be getting 0.96, which is almost equal to 1, 1.0. Now actually; that means, if I just try all the bits then on an average probabilistically there will be mutation only on 1 bit. So, out of these 96 the mutation will occur only on 1 bit.

Now, how to implement? Now to implement that, actually what you will have to do is. So, you will have to concentrate on this particular the children solution, you concentrate on the first bit position and once again, you try to implement this probability of this mutation, with the help of the random number generator.

So, the random number generator is going to generate a number lying between 0 and 1. Now, if it can generates a number which is less than equals to 0.01, then that will be a success. So, there will be a mutation and if there is a failure. So, there will be no mutation and it will remain same and it will remain intact, the bit will remain intact. Now, bitwise I will have to check whether it is going to participate in mutation or not.

So, positive concentrate on the first bit, second bit, third bit and you just go on trying on all the bits; that means, all 96 bits and as I told on an average, there could be mutation only on 1 bit. Now, supposing that fortunately or unfortunately. So, this particular bit at this particular position it is the decision has been taken, that there will be some mutation and once it has been decided, there will be some mutation. So, this particular 0 will become 1 and if there is a 1 that will become 0, this particular 0 will be converted to 1 through this particular the bitwise mutation.

(Refer Slide Time: 17:24)



Now, if I just implement the bitwise mutation, then I will be getting this particular modified population of solution, after the mutation and we can see that here. So, this previous 0, that has been converted to 1 and this is actually the population of solution, after the mutation is over. So, this is the population of solution, which will be getting after the mutation and once I am got this particular thing, then we try to find out what should be the fitness information.

So, corresponding to the first ga string the same procedure you will have to follow, the first 7 is going to represent b. So, 1 2 3 4 5 6 7. So, these are going to represent b, the remaining 9 is going to represent h, and we will have to find out the decoded value, the real value for b and h. So, this is b and this is going to represent h, and once you have got the real values for b and h, I can find out what should be the value of the objective function and if we calculate.

So, it will be getting the values of the objective function or the fitness for all this solution. So, for the first ga string after mutation, this will be the fitness for the second one, this is the fitness 0.57 third one 5.03 the 4th one 2.64 fifth one 10.58 and the 6th one 1.48 and once you have got this particular fitness information, we will have to check once again, whether there is any violation of the functional constraint.

Now, here fortunately there is no violation of the functional constraint. So, all the solutions are feasible and if I just compare, the fitness values for this second one we are getting the minimum fitness there is 0.57 and this is a minimization problem. So, at the end of the first iteration or at the end of the first generation of the g a, this is actually the best solution and this is also a feasible solution.

Now, if I compare the best feasible solution, whatever we got in this particular the initial population that was 0.85, that was best feasible solution, in the initial population of the ga and at the end of this particular iteration, I am getting 0.57 as the best solution and this is one second found to be a feasible solution. So, from 0.85 the fitness value is going to be reduced to 0.57. So, even in 1 iteration or in 1 generation, there is significant improvement in the value of this particular the fitness.

So, in one iteration ga can improve the solution and this process will go on and go on. Through a large number of iteration or the generation, pre specified by the user and ga will be able to hit, the optimal values for the b and h; that means, it will be able to find out, the optimal cross section for this particular the single point cutting tool and for this single point cutting tool, if it can find out that particular your the cross section, the length is fixed and I can also find out, that complete optimal design of this particular, your the single point cutting tool and this single point cutting tool will be able to perform the turning operation, on length without any mechanical breakage.

Thank you.