

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

Lecture - 07
Binary – Coded Generic Algorithm (BCGA) (Contd.)

Now, I am going to start with a numerical example to show you how does a binary coded GA can optimize one objective function. So, I am just going to take one example.

(Refer Slide Time: 00:34)

Hand Calculation

Maximize $y = \sqrt{x}$

subject to $1.0 \leq x \leq 16.0$

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now supposing that I will have to solve one maximization problem of this particular form y is a function of x that is a y equals to square root of x . So, this is nothing, but a function of only one variable. Subject to the condition that x is lying between 1 and 16 and I am just going to consider that x is nothing, but a real variable.

Now, very easily we can find out the solution of this particular problem. Now, the maximum value of x is 16.0 and square root of that, so the solution is 4. So, the maximum value of y is nothing, but 4. Now, let us see how does a binary coded GA can improve its solution iteration wise just to reach that the globally maximum solution that is nothing, but 4.0.

(Refer Slide Time: 01:40)

Hand Calculation (contd.)

String No.	Initial population	Decoded value	x value	$f(x)=\sqrt{x}$	$p_{selection} = \frac{f_i}{\sum f}$	Expected count $\frac{f_i}{\bar{f}}$
1	100101	37	9.81	3.13	0.18	1.07
2	011010	26	7.19	2.68	0.15	0.91
3	010110	22	6.24	2.50	0.14	0.85
4	111010	58	14.81	3.85	0.22	1.31
5	101100	44	11.48	3.39	0.19	1.16
6	001101	13	4.09	2.02	0.12	0.69
				sum $\sum f = 17.57$		
				average $\bar{f} = 2.93$		
				maximum $f = 3.85$		

Handwritten notes in red:
 $N = 6 \times 6 = 36$
 $100101 = 32 + 4 + 1 = 37$




Now, here actually what you do is if you remember the GA the binary coded GA starts with a population of solution selected at random. Now, here I am just going to consider what simplicity that the population size is equal to 6; that means, your N is equals to 6 and I am just going to consider only 6 bits to represent the variable that is your 1 2 3 4 5 6. So, 6 multiplied by 6, so 36 bits are generated at random initially using the random number generator.

Now, supposing that, so this particular initial population of solution, we have generated at random. Now, how to find out the optimal solution how a binary coded GA can find out the optimal solution. Now, let us try to concentrate on the first GA string that is a 1 0 0 1 0 1. So, will have to find out the decoded value corresponding to this particular you are the binary.

Now, if you want to find out the decoded value that is 1 0 0 1 0 1, so this is 0. So, the decoded value its place value is 2 raise to the power 0, 2 raise to the power 1, 2 raise to the power 2, 2 raise to the power 3, 2 raise to the power 4, 2 raise to the power 5. So, this is nothing but 32, 2 raise to the power 5 is 32 plus 2 raise to the power 2 is 4 plus 2 raise to the power 0 is nothing, but 1, so 32 plus 4 plus 1, I will be getting the decoded value as 37.

Now, what I do is, will be getting that particular decoded value as 37. Similarly we can find out the decoded value for the second GA string that is 26, for the third one is 22, 4th one is 58, fifth one 44 and your 6th one is nothing, but 13.

(Refer Slide Time: 04:00)

Hand Calculation (contd.)

String No.	Initial population	Decoded value	x value	$f(x)=\sqrt{x}$	$p_{selection} = \frac{f_i}{\sum f}$	Expected count $\frac{f_i}{f}$
1	100101	37	9.81	3.13	0.18	1.07
2	011010	26	7.19	2.68	0.15	0.91
3	010110	22	6.24	2.50	0.14	0.85
4	111010	58	14.81	3.85	0.22	1.31
5	101100	44	11.48	3.39	0.19	1.16
6	001101	13	4.09	2.02	0.12	0.69
				sum $\sum f = 17.57$	$\sum = 1.0$	
				average $f = 2.93$		
				maximum $f = 3.85$		

Handwritten notes on the slide:
 - A red circle around the first string (100101) with 'x' written next to it.
 - A red circle around the first row of the table.
 - A red circle around the 'x' value column header.
 - A red circle around the 'f(x)=sqrt(x)' column header.
 - A red circle around the 'p_selection' column header.
 - A red circle around the 'Expected count' column header.
 - A red circle around the 'sum sum f = 17.57' calculation.
 - A red circle around the 'average f = 2.93' calculation.
 - A red circle around the 'maximum f = 3.85' calculation.
 - A red circle around the 'sum = 1.0' calculation.
 - A red circle around the 'x' value for the first string (9.81).
 - A red circle around the 'f(x)' value for the first string (3.13).
 - A red circle around the 'p_selection' value for the first string (0.18).
 - A red circle around the 'Expected count' value for the first string (1.07).
 - A red circle around the 'x' value for the second string (7.19).
 - A red circle around the 'f(x)' value for the second string (2.68).
 - A red circle around the 'p_selection' value for the second string (0.15).
 - A red circle around the 'Expected count' value for the second string (0.91).
 - A red circle around the 'x' value for the third string (6.24).
 - A red circle around the 'f(x)' value for the third string (2.50).
 - A red circle around the 'p_selection' value for the third string (0.14).
 - A red circle around the 'Expected count' value for the third string (0.85).
 - A red circle around the 'x' value for the fourth string (14.81).
 - A red circle around the 'f(x)' value for the fourth string (3.85).
 - A red circle around the 'p_selection' value for the fourth string (0.22).
 - A red circle around the 'Expected count' value for the fourth string (1.31).
 - A red circle around the 'x' value for the fifth string (11.48).
 - A red circle around the 'f(x)' value for the fifth string (3.39).
 - A red circle around the 'p_selection' value for the fifth string (0.19).
 - A red circle around the 'Expected count' value for the fifth string (1.16).
 - A red circle around the 'x' value for the sixth string (4.09).
 - A red circle around the 'f(x)' value for the sixth string (2.02).
 - A red circle around the 'p_selection' value for the sixth string (0.12).
 - A red circle around the 'Expected count' value for the sixth string (0.69).
 - A red circle around the 'sum sum f = 17.57' calculation.
 - A red circle around the 'average f = 2.93' calculation.
 - A red circle around the 'maximum f = 3.85' calculation.
 - A red circle around the 'sum = 1.0' calculation.

And once we have got the decoded value for each of the GA string. Now, I will have to find out the real value corresponding to this decoded one and to find out the real value actually we use the linear mapping rule. The linear mapping rule is as follows like your x real value is nothing, but $x_{minimum}$ $x_{minimum}$ plus $x_{maximum}$ minus $x_{minimum}$ divided by 2 raised to the power 1 minus 1 multiplied by the decoded value.

Now, here $x_{minimum}$ is nothing, but 1.0 $x_{r_{maximum}}$ is 16.0. So, 16.0 minus 1 point 0 is 15.0 divided by 2 raise to the power 1 is equals to 6 minus 1 into the decoded value. And if you follow this particular linear mapping rule. So, will be getting the decoded value for the x the real value for the x for the first string will be getting 9.81, the second one 7.19, the third one 6.24, fourth one 14.81, fifth one 11.48 and the sixth one 4.09. And once I have got the real values of the variable now, you substitute here and he will be getting the value of the objective function and this is a maximization problem. So, we can find out the fitness values. Now, here the fitness value is square root of 9.81 that is nothing, but 3.13.

Next is square root of 7.19 is 2.68 and so on for the last one the square root of 4.09 is nothing, but 2.02. Now, this is a maximization problem. So, these are nothing, but all

fitness values. Now, I can find out the fitness values of all the solutions I can find out what should be the sum of all fitness values and that is coming as 17.57 and I can also find out the average that is nothing, but 17.57 divided by 6 and this will become equal to 2.93. And if we just look into this fitness information I can find out that out of all the fitness values the maximum is 3.85. So, the maximum fitness is 3.85.

And once I got this particular the fitness information, now I am just going to start with the proportionate selection that is nothing, but the roulette wheel selection. Now, according to the proportionate selection the probability of selection is nothing, but the individual fitness individual fitness divided by the sum of the all fitness values. So, for the first string thing this will become equal to 0.18, second one 0.15, third one 0.14, fourth one 0.22, fifth one 0.19, sixth one 0.12. Now, if you just find out the sum of all the probability values. So, this will become equal to 1.0. Now, we can find out once I have got this particular probability value. Now, we are in a position to find out what should be the expected count. By expected count we mean how many times a particular bit will be copied in the mating pool through this particular the reproduction scheme.

Now, this expected count is calculated as a f_i divided by \bar{f} \bar{f} is nothing, but the average fitness. So, this is the average fitness. Now, I can find out, so f_i divided by \bar{f} that is 3.13 divided by 2.93 is nothing, but 1.07. Similarly for the second string it is 2.68 divided by 2.93 and this will become equal to 0.91. Similarly for all the GA string I can find out what should be your the expected count.

Now, as I told by expected count we mean how many times a particular GA string will be copied in the mating pool. Now, the actual count could be slightly different from this particular the expected count. Now, let me try to find out what could be the expected count and this is actually probabilistic.

(Refer Slide Time: 09:30)

Hand Calculation (contd.)

Actual count Roulette wheel	Mating pool	Mating pair	Parents	Crossover site	Children strings	Mutation
1	100101	3	100101	10 0101	101010	101010
1	011010	6	111010	11 1010	110101	110101
0	111010	1	011010	011 010	011100	011100
2	111010	5	101100	101 100	101010	111010
2	101100	4	111010	11 1010	111100	111100
0	101100	2	101100	10 1100	101010	101010

$P_c = 1.0$
 $P_m = 0.03$
 $36 \times 0.03 = 1.08$

So, what you can do is we can find out this actual count. So, might be actual count for the first string is 1, the second string is 1, third string is 0; that means, the third string will not be copied in the mating pool for the 4 string it is 2 that mean the 4 string will be copied twice in the mating pool. The fifth one the actual count is 2; that means, the fifth string will be copied twice in the mating pool and for the 6 string it is 0; that means, it will not be copied.

Now, if I copy the first string only ones. So, this is actually the first string 1 0 0 1 0 1, 1 0 0 1 0 1. So, 1 0 0 1 0 1 will be copied only ones. The second string the actual count is 1. Now, if you see the second string 0 1 1 0 1 0. So, this will be copied only ones. The third one is not copied in the mating pool the fourth one that is this particular thing 1 1 1 0 1 0. So, that will be copied twice 1 1 1 0 1 0, 1 1 1 0 1 0 next is the fifth string is 1 0 1 1 0 0 and this will be copied twice 1 0 1 1 0 0, 1 0 1 1 0 0. So, this is copied twice. So, this is the way actually using the proportionate selection or the roulette wheel selection. So, we will be getting this particular the mating pool.

Now, if I compare the average fitness of this particular mating pool with the average fitness of the initial population there is a possibility that the average fitness of the mating pool will be higher compared to that of the initial population, because in the mating pool there could be multiple copies of the good solution and the bad solutions will be deleted from the mating pool that is the purpose of reproduction scheme. And once I have got

this particular the mating pool. Now, I can select the mating pair and the reason why you go for the mating pool we want to select only the good solutions because if the parent solutions are good the probability that will be getting the children solution will also be more and that is why in the mating pool we actually tried to select the good solutions only and there could be multiple copies of the good solutions. Now, once we have got this particular mating pool.

Now, let us see how to find out the mating pair. Now, mating pair is actually a selected at random. Now, let me concentrate on the first one let me concentrate on the first your the first GA string and let us see how to select the mating pair. Now, this is actually the first GA string of the mating pool and I will have to find out its mating pair. Now, this particular GA string is going to meet with the third one and that is selected at random using the random number generator. So, this is going to meet with the third one. So, here I have written 1 3; that means, 1 is going to meet with 3; that means, 3 is also going to meet with 1.

Next is I will concentrate on the second GA string say the second GA string is going to meet with the sixth one. So, this is going to meet with the sixth one. So, 2 is going to meet with sixth, six is going to meet with 2 and accordingly I will be getting the parents that I will be discussing after sometime. Now, let me concentrate on this particular the fourth one. The fourth is going to meet with the fifth and fifth is also going to meet with the 4.

Now, let us see how to get the parents the mating parents. Now, as I told 1 is going to meet with 3. So, let me copy 1 here that is 1 0 0 1 0 1, 1 0 0 1 0 1. Now, I will have to copy 3 1 1 1, 1 1 1 0 1 0, 1 1 1 0 1 0. So, these 2 will be the first mating pair the next is your the second one that is 0 1 1 0 1 0, 0 1 1 0 1 0 and it is going to meet with the sixth that is 1 0 1 1 0 0, 1 0 1 1 0 0. So, they are going to meet together next is your this particular solution is the fourth, fourth is going to meet with 5 and forth and copying it here 1 1 1 0 1 0, 1 1 1 0 1 0 and the fifth one is nothing, but 1 0 1 1 0 0, 1 0 1 1 0 0. So, these 2 parents are going to meet. Now, I have got the mating pairs.

The population size is 6, so there will be 3 mating pairs because each pair consists of 2 solutions. Now, supposing that the probability of mutation sorry the probability of crossover that is p_c is equals to 1.0; that means, all 3 mating pairs are you going to

participate in cross over and let me also assume that that probability of mutation say it is equal to say 0.03. So, it is very hypothetical value. Now, let us see how to proceed with the crossover and how to proceed with the mutation to get the final solution at the end of the first iteration.

Now, let me once again concentrate on the crossover and here we are going to consider the single point crossover. Now, it is having 5 6 bits; that means, there will be 5 places for crossover, so 1 2 3 sorry 1 2 3 4 5. So, there are 5 places for crossover. So, out of this 5 sides anyone will be selected at random using the random number generator. Supposing that, this particular site is selected as the crossover site.

Now, if this is the crossover site. So, you are the parents are 1 0 0 1 0 1, 1 1 1 0 1 0 and this is the crossover site. So, on left hand side there will be no change; that means, on the children solution I will be getting 1 0 1 1, 1 0 1 1 and on the right hand side. So, there will be swapping, whatever is there at the bottom that will go to the top and vice versa. So, I will be getting this 1 0 1 0 here, 1 0 1 0 here and 0 1 0 1 here, 0 1 0 1 here. So, using the single point crossover from 2 parents I can find out 2 children solution. Similarly for the second mating pair, this is the crossover site and I will be getting the children solution like this. For the third parent, this is the crossover site and using the single point crossover. So, I will be getting 2 children like this and this. So, I am getting the children solution and here there will be diversification of the properties due to the exchange of properties.

Now, if the parents are good the children solutions are expected to be good or slightly better, but of course, there is no guarantee. Now, once I have got this particular children solution. Now, we just go for the mutation. Now, as we assume that the mutation probability equals to 0.03 and we have got how many bits 6 multiplied by 6 the population size is 6 each GA string is having 6 bits, so we have got 36 bits. So, 36 multiplied by 0.03 and that is nothing, but 1.08; that means, on an average only on 1 bit there will be mutation. But there is no guarantee there could be no mutation also there could be mutation on 2 bits also, but let me assume that mutation will occur only on 1 bit and what will have to do is at each of the bit position. So, I will have to use the probability that is 0.03 of appearing head in coin tossing and if head appears it will be a success and there will be actually the change of bits. Otherwise it will remain the same.

So, at each of the bit position we try with the probability of appearing head in coin tossing that is 0.03 and there is the possibility that out of this 36 bits the mutation will occur or 1 bit and supposing that the mutation is coming here; that means, this particular 0 that will be converted to 1; that means, using the mutation. So, I will be getting, this type of solution and other bits other string will remain the same. So, this is what is happening at the end of this particular the mutation and that completes in fact, one cycle of this particular the GA. And let us see what happens at the end of this particular the iteration.

Now, if you see what is going to happen at the end of this, so once again the population which you are getting we try to find out the decoded value using the method which have already discussed.

(Refer Slide Time: 20:24)

Hand Calculation (contd.)

Decoded value	x value	$f(x) = \sqrt{x}$
42	11.00	3.32
53	13.62	3.69
28	7.67	2.77
58	14.81	3.85
60	15.28	3.91
42	11.00	3.32
		sum $\sum f = 20.86$
		average $\bar{f} = 3.48$
		maximum $f = 3.91$



38

And once I have got the decoded value now, I can find out the real values using the linear mapping rule the why I discussed. And once I have got the real values now, I can find out the function value and which is nothing, but the fitness values. And if this is the fitness values, now I can find out the sum of all the fitness values is 2 point sorry 20.86 and the average fitness that is the total fitness divided by 6 is 3.48 and if with the search what is the maximum fitness you can find out 3.91 is found to be the maximum.

Now, if you correctly remember in the previous or initial population we had the maximum fitness was your 3.85 and. Now, it has become 3.91. So, the maximum fitness

is going to increase and what is the globally maximum solution that is 4.0. So, in one generation or one iteration the GA is moving towards that globally optimal solution that is 4.0 and it has shifted little bit towards 4 that means from 3.85 it has moved up to 3.91. So, this completes one iteration of GA. Now, this particular process will go on and go on. So, it will run this GA for a large number of iteration and then this particular the binary coded GA is going to hit that globally maximum solution for this particular function that is 4.0. Now, this is the way actually a binary coded GA works.

Now, this particular numerical example gives some confidence that yes the GA can solve the optimization problem it can find out the globally optimal solution and it gives actually that type of confidence. Now, I am just going to discuss or like how to set the parameters in GA. Now, one thing we should we should understand that if you want to run GA will have to set the GA parameters properly.

(Refer Slide Time: 23:19)

GA- parameters Setting

- To ensure a proper balance between **population diversity (exploration)** and **selection pressure (exploitation)**

✓
 p_c ✓
 p_m ✓
 N ✓
 G_{max} ✓

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Now, by the parameters we mean actually what should be the value for this particular the crossover probability that is p_c , then comes the mutation probability that is p_m , then comes the population size that is N , then maximum number of generation that is your the G_{max} .

Now, let us see what happens if I take a very high value or a very low value for each of this particular the parameters. Now, if I take a high value denoted by H for this particular p_c what will happen. Now, the high value means it is very say 0.9 or say 1 , if it is 1

then all the mating pairs are going to participate in cross over. There will be exchange of properties and the diversity is bound to come. On the other hand if I consider a low value for this particular the probability of crossover the very purpose of using the crossover operator may not be soft, all the mating pairs will not be allowed to participate in crossover and we may not get the desired diversity in the children solution.

Now, let us see what happens if I consider a very high value for this particular p_m the probability of mutation. Now, if I take a very high value for this particular p_m . So, this GA search will become a random search. On the other hand if I take a low value for this p_m the very purpose of using the mutation operator may not be soft and it may not be able to help to come out of the local minima problem.

Now, let us see what happens if I take a very high value for this particular the population size. If I take the large population size the GA search will be proper, but at the cost of more computation. On the other hand if I take a small value for this particular end the search will not be proper and there will be a chance of premature convergence. Now, let us see what happens if I take a very high value for this particular the G_{max} that is the maximum number of generation. If I take a high value for this particular G_{max} there is a possibility will be getting very good search, but once again at the cost of more computation. On the other hand if I take very low value for this particular G_{max} the search may not be proper and there is a chance that he will be getting some sort of premature convergence.

Now, if I want to use GA to solve some optimization problem. So, I will have to find out the optimal values of this particular the GA parameters. Now, how to determine this optimal GA parameters? Different, there are different ways. Now, here I am just going to discuss a particular method in details. But before that let me tell you the purpose of finding the optimal GA parameters is to ensure a proper balance between the population diversity and the selection pressure.

Now, I have already discussed the importance of maintaining a proper balance of population diversity and selection pressure. So, we have understood that if I want to ensure a very efficient search for the GA a proper balance between population diversity and selection pressure has to be maintained. And how to maintain the balance to maintain the balance will have to use the proper the set of this particular the GA parameters. So,

we will have to use optimal GA parameters. Now, how to find out the optimal GA parameters that I will be discussing.

Thank you.