Design and Optimization of Energy Systems
Prof. C. Balaji
Department of Mechanical Engineering
Indian Institute of Technology, Madras

Module No. # 01
Lecture No. # 39
Genetic Algorithms Contd…

So, we will continue with the discussion on genetic algorithms. So, the first part of the last lecture, we saw the logic behind developing the genetic algorithm. So, GA is a essentially evolutionary optimization technique, techniques, where we try to mimic the process of evolution.

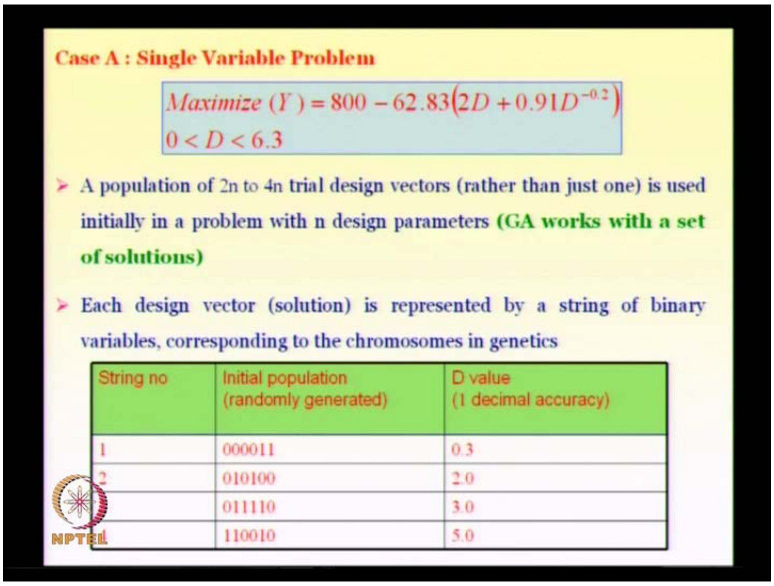
Evolution is stated as an optimizing process, and then you do not deal with, you do not deal directly with the variables; you deal with a coded set of variables, that is, the variables that are converted into 0s and 1s. Some people may ask, why it is required, cannot we do without this? It is possible, yes; suppose if there are 2 design vectors, the 2 variables x 1 and x 2, it is possible to combine them with, 1 minus lambda into x 1 plus lambda into x 2.

And, lambda can be decided based on tossing a coin or based on random number table and so on, but it is easier to represent it as a binary. Because, you can have many number of bits, and lot of combinations for crossover, mutation, exchange, so the mating, reproduction, all those things; that means, getting new design vectors from the old ones becomes easier this way. Therefore, the traditional genetic algorithm is based on the binary representation; of course, lots of representations are there. Towards the end of the lecture I told, you know, there are lots of representations for mutation, crossover.

So, generally, when you look at some algorithm, you do not know whether it is genetic algorithm or not, because you can customize it to suit your needs. But, basically there should be some stochastic elements in this, that is certain things are randomly decided. In our problem, the mate was randomly decided, the place where the cross over took place was randomly decided, so there are some stochastic or probabilistic rules. But, there are some deterministic rules like, the fittest alone will survive, the fittest alone will reproduce.

So, you are actually putting your fundas of maximizing your y or minimizing your y, but you are, by using stochastic principles, you are ensuring that you do not get into premature convergence, you do not get into a local optimum or minimum. Therefore, GA essentially strives to attain the global maximum or minimum.

(Refer Slide Time: 02:23)



Since, evolution is generally a maximizing process, where the fitness of the population increases, the original GA is applied to a maximization problem. That is why I put, 800 minus 62.83, but it is possible for you to treat it to solve the minimization problem by choosing that species, which has the lowest fitness, and giving it maximum priority for the mating and reproduction; are you getting the point?

So, you can essentially use, directly use, x 1, x 2, x 3, x 4, for example, string 1, string 2, string 3, string 4, if you are interested in the maximization, you will rank them in the descending order of their fitness. You can also rank them in the descending order of the fitness, and choose that which has the least fitness, if you are seeking a minimum to the problem; it will become clear, when we take up an example in today's class.

So, I will just go through the example which we did in the last class. So, this is the heat transfer from a spherical reactor, so we converted into one variable problem, and Q is

62.83 into 2 D plus 0.9 D to the power of minus 0.2, the first term is D to the power of 1, the second term is D to the power of minus 0.2. Therefore, there is hope that there should be a particular value of D, at which it reaches an extremum, it becomes stationary, either a maximum or minimum.

Now, we are seeking a minimum to the problem; therefore, we are trying to apply the traditional GA. So, we try to maximize 800 minus 62.83. This is not the only way of doing it; you could have maximized 1 by Q, you could have just minimized Q, but you have to take care when you are doing the various operations. I chose 800, so that you get a positive value always; yes.

Student: Excuse me sir, we can just put minus 1.

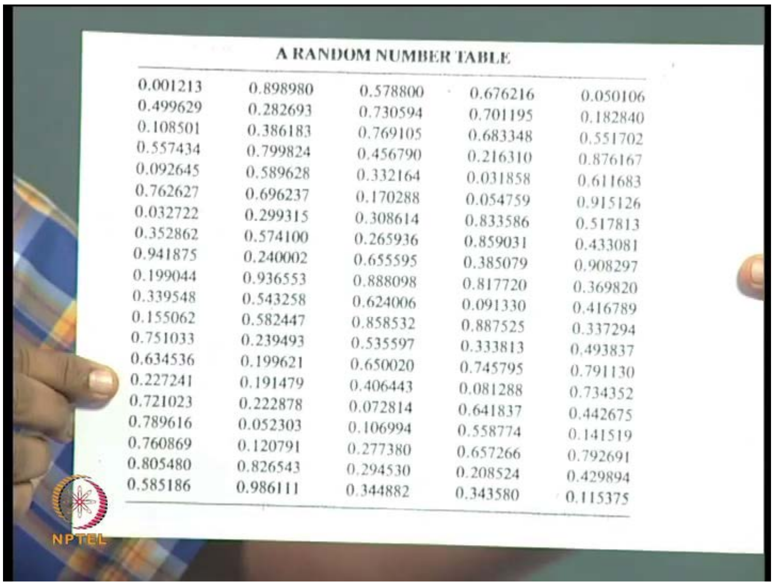
Yes, I want it, I took 800, so that for all possible values of D in our range, that is 0 less than x, less than equal to 6.3 meters; I cannot apply 0 itself, because D to the power of minus 0.2 is there. Therefore, I cannot apply 0, so you cannot have 0 diameters that is a crucial solution, but I ensure that D maximum is 6.3, so that 6 bits are enough, they are sufficient.

So, there are 4 string, so we took 0.3, 2.0, 3.0 and 5.0. So, what I will do is, in today's class I will give you the random number table, may be I will give it away; Alok, you just give it to; how do you show it to the others? You want to take a close up of this, how do you show it to the others? What you know?

Student: Voice not clear.

Yes, how? So, you will listen.

(Refer Slide Time: 05:24)



A RANDOM NUMBER TABLE

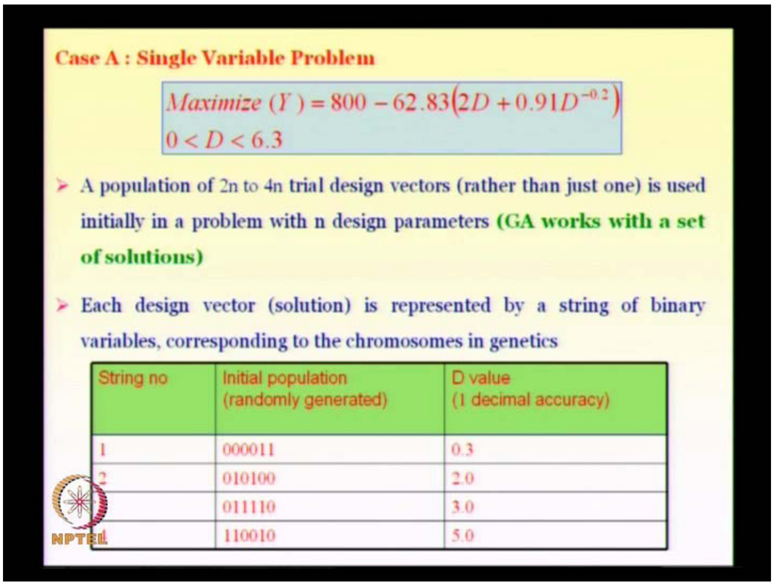
0.001213	0.898980	0.578800	0.676216	0.050106
0.499629	0.282693	0.730594	0.701195	0.182840
0.108501	0.386183	0.769105	0.683348	0.551702
0.557434	0.799824	0.456790	0.216310	0.876167
0.092645	0.589628	0.332164	0.031858	0.611683
0.762627	0.696237	0.170288	0.054759	0.915126
0.032722	0.299315	0.308614	0.833586	0.517813
0.352862	0.574100	0.265936	0.859031	0.433081
0.941875	0.240002	0.655595	0.385079	0.908297
0.199044	0.936553	0.888098	0.817720	0.369820
0.339548	0.543258	0.624006	0.091330	0.416789
0.155062	0.582447	0.858532	0.887525	0.337294
0.751033	0.239493	0.535597	0.333813	0.493837
0.634536	0.199621	0.650020	0.745795	0.791130
0.227241	0.191479	0.406443	0.081288	0.734352
0.721023	0.222878	0.072814	0.641837	0.442675
0.789616	0.052303	0.106994	0.558774	0.141519
0.760869	0.120791	0.277380	0.657266	0.792691
0.805480	0.826543	0.294530	0.208524	0.429894
0.585186	0.986111	0.344882	0.343580	0.115375

Do it like this. I could have made a ppt of this thing, right? I hope all of you have the random number table. So, it is possible. You can use this in the exam. Knowing you fully well, I know that many of you will not bring it to the exam, so I will give 1 more copy in the exam, but try to bring this to the exam, or I will integrate it with the question paper, but this is suppose if you want to practice some problems or you can use this, everybody has, you have; Vinith, you have?

So, please look at the random number table. Student: We do not have. You do not have, how it is? Sampath, is it over, I got 85 copies.

In MAT lab or in a computer program, you can just put rand of x, r a n d of x, it will generate a random number. This random number is all positive between 0 and 1, and you should go from top left, the sequence should be like this, when you are using it in genetic algorithms, or when, if you have learnt probability the teachers would have told you that you have to use a random numbers in this sequence, you should not break, you start from the left, go down and go down, in a serpentine manner.

(Refer Slide Time: 07:11)

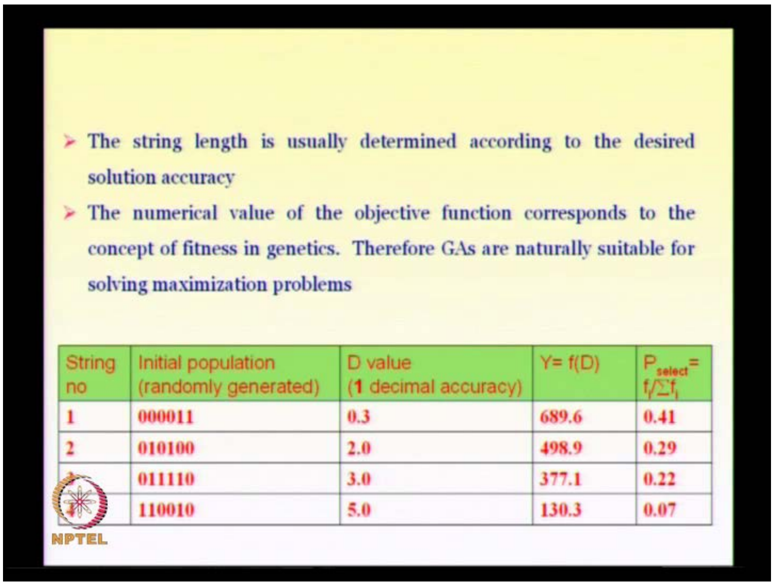


So, what you can do is, so if you do not want to do this 0.3, if you do not want to do 0.3, 2.0, 3.0 and 5.0, you want to randomly select; that means, you want 6 6, you want to select 6 bits, for every value of the diameter. So, we can say that, generate a random number. If it is greater than 0.5, the bit is 1; if it is less than 0.5, the bit is 0; and proceed in this order, so you will get 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1; are you getting that? And, you will fill up like this, as you look at the table, you will fill up. So, in the exam you are encouraged to do this.

The other way of doing it would be, here I told you it is 0 and less than 6.3, so you take 4 values in between, so I took 0.3, 2.0, 3.0 and 5.0. What is the test that it is really random? After you have generated, if it is, if all these 4 are equally spread in the original interval of uncertainty, the number of 0s will be more or less equal to the number of 1s. But, if you generate it using random number, it will be exactly the same, within a small, may be as a slight difference, that is all, is it ok. So, depending on whichever is convenient, this is the threateningly formal way of doing it; bit by bit you generate by using the random number table; we have to go from this to this, alright.

Now, if you see, did I make the number of 0s equal to number of 1s? If it is 24 bits, if you have 11, 13 or 13, 11, it is ok, but you should not have 8 and 16. That means, your solution is biased, either towards low values of D or high values of D, which means there is a chance that you will get good convergence, which is premature convergence. Please remember the idea of, the idea of developing an algorithm like GA, is to overcome this premature convergence, and overcome this local minima or maxima. Therefore, your original solution should be widespread, should be spread throughout the interval, so that there is no chance of missing the global minimum or maximum, is that ok? Is this step clear now? You can generate either 0s or 1s using this, or you just randomly put 0.3, 2.0, 3.0 and 5.0.

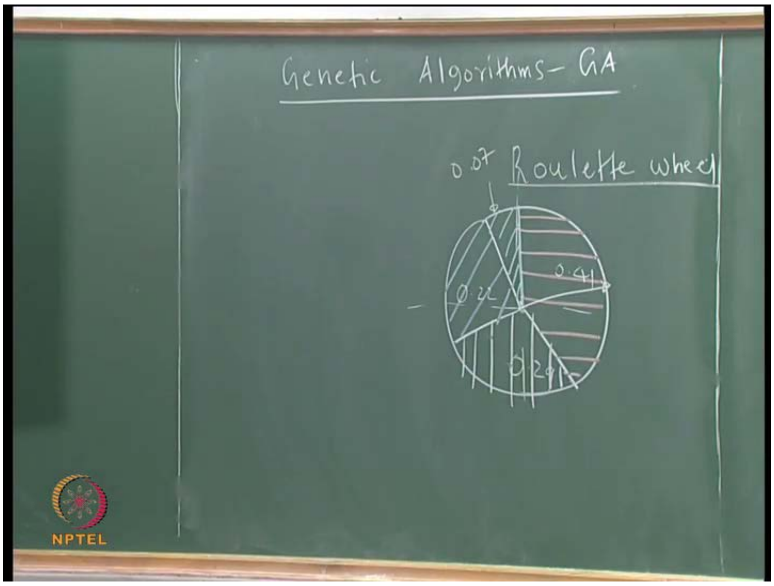
(Refer Slide Time: 09:33)



Next step, now, this, I know the value of D, so, apply; so I use the algorithm, I use the formula, 800 minus 62.83, substitute the value of D and get the value of Y; Y is 800 minus Q; I want to maximize Y, right. So, I generate the 4 values of Y, for the 4 values of D.

Now, what I do is, I add all the values of Y, so 700 plus 500, 1200, 1600 about 1750 or 1800; then, I divide 689.6 divided by 1800, 498 divided by 1800 and so on, and get the relative fitness, so that the sum of all this equal to 1.

(Refer Slide Time: 10:10)



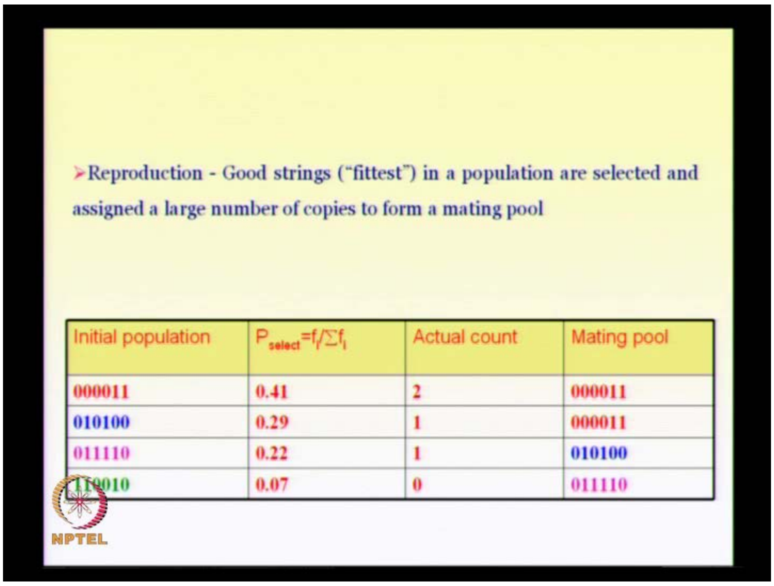
What is the idea behind this? You can imagine what is called a, I am not sure about the spelling, Roulette wheel.

Now, suppose you take a circle, and then, so this is 25 percent, so 37.5, 41, 70, so, this 0.41. Suppose, you make a wheel like this, let it be made of cardboard or whatever, you make a hole in the center, then you have a pointer, you have a pointer, allow it to rotate and ensure that it comes to rest. Do this many number of times; if you do it for 200 times, there is a 41 percent chance, that it will be here. So, there will be a 29 percent chance, that it will lie in this region, there is 22 percent chance.

Therefore, this; so, the generation of parents, that is you choose among the fittest in the population is analogous to; that is this p select know 0.41, 0.29, 0.22, 0.07 is analogous to finding the number of times a pointer will go to that particular region in a Roulette wheel, is that ok? But now since it is only 4 strings, it is easy for us to manually choose.

So, since 41 percent is very strong, I am seeking maximization; this is the maximum. I want 2 times this, in the next generation. So, I will take 2 times this, 1 and 1. Please remember, this is what is called a constant population genetic algorithm, there are other types of genetic algorithm also, this is the constant population genetic algorithm. That means, each generation or iteration, I want only 4 members of the population. So, I will choose 2 times of this first one, 1 time this one, 1 time this one, this has got very poor fitness, it gets removed.

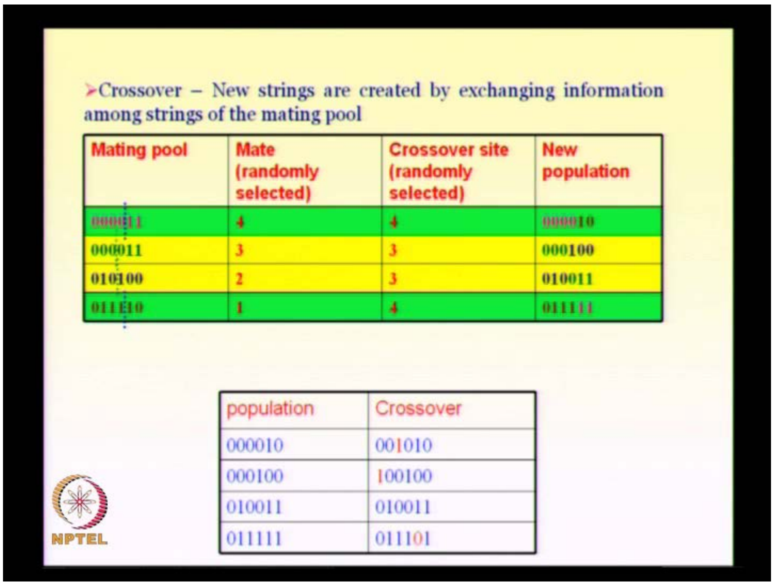
(Refer Slide Time: 12:47)



So, now, I am choosing this. So, this 0.41, the first one is red color, the other is purple, the other is pink and the 4th is green. Now, the actual count is 2, 1, 1, 0, therefore the green one is completely eliminated from the mating pool; this is called the mating pool. That means, from these 4, we will produce 4 children, which will have a combination of the bits, which are present in these 4 candidates, is this clear?

So, already, we are actually looking at Y value, for which of these candidates is Y more? Therefore, wherever Y is more, we are giving more preference to that; that is where the evolution is being used; that is, something which has a better fitness, has got a better chance to survive and reproduce.

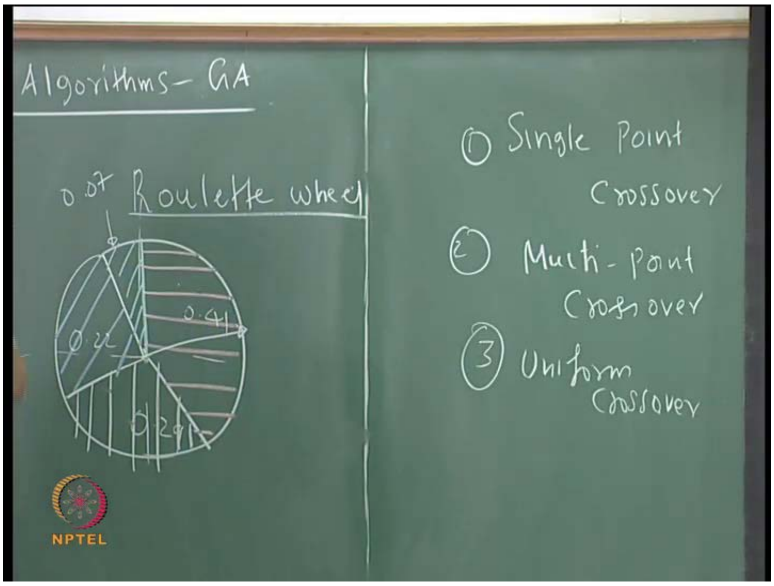
(Refer Slide Time: 13:37)



Now, this pink one, 00011, 00011, is essentially the first one, then I put the second one and third one, these are the mating, this is the mating pool. So, we can play around with this 24 bits. So, we have to choose the mate. For 1 and 2 are essentially the same, therefore, 1; there is no point in putting 2 as the mate for this and 1 as the mate for this. So, for this you have to put either 3 or 4, I am choosing 4; if this becomes 4, for the 4th one, the mate will be 1; are you getting the point? So, needless to say, the mate for this will be 3 and the mate for this will be 2, now this is randomly decided. You can still take a, toss a coin and write your algorithm and put this.

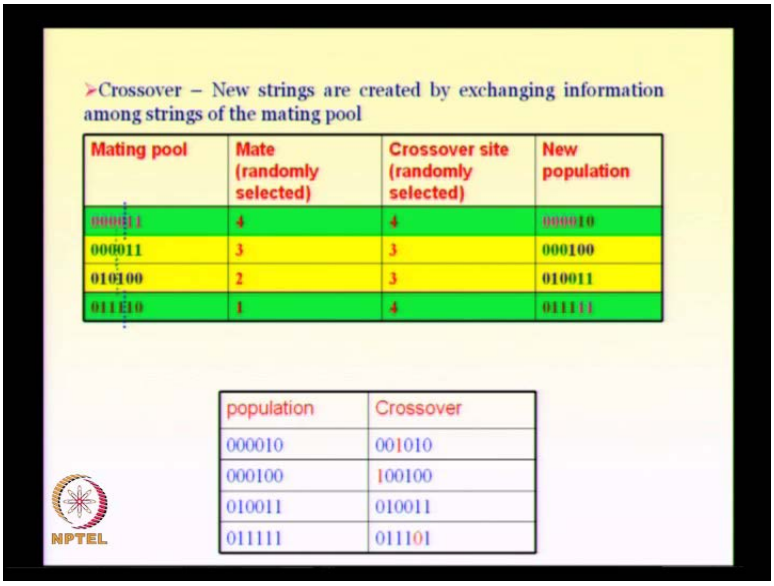
Now, you have to have the crossover site. The crossover site is the point at which you want to divide the chromosome and exchange the bits. Suppose, I put the crossover site is 4, then I have 0000; that is, at the end of the 4th bit, I want the change; therefore, the new population will be 000010, right, 000010. So, the other one will be 0111, because its mate is 4, 011111. So, for, this is what is called as a single point crossover; you can also have multi point crossover.

(Refer Slide Time: 15:31)



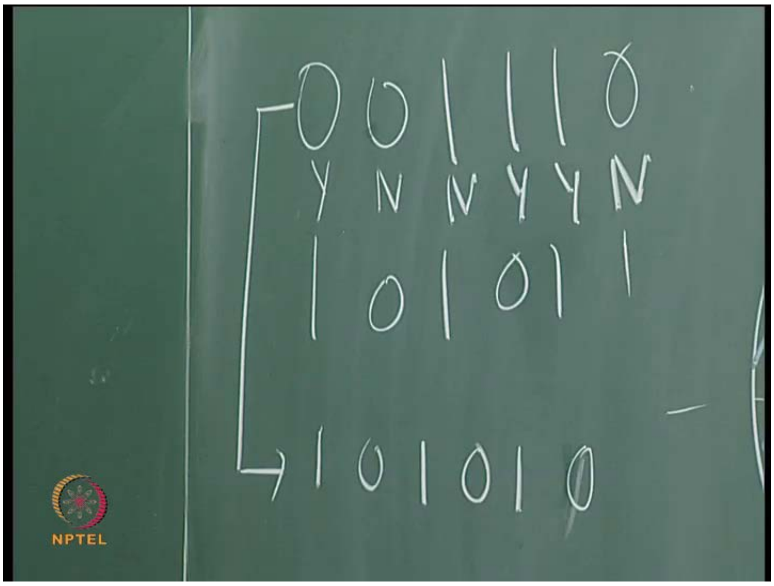
What is multi point crossover? You can have crossover at the end of the 2nd as well as the 5th bit or 2nd and 4th bit; or, you can have uniform crossover. So, in the exam, you can do whatever you want, but you tell, you write clearly what you, what you are doing. What is uniform crossover? I will tell you what, suppose you are looking at these 2, a uniform crossover, please look at the board very carefully; if you have already exhausted 16 random numbers, you have to put a, you have to put a marking on the random number table with your pen or pencil, when you are using it in the exam.

(Refer Slide Time: 15:50)



So, you have exhausted the first 16 or 24 random numbers for generating the bits. Go to the 25th bit, if the random number is greater than 0.5, you will say crossover, yes; if the random number is less than 0.5, the crossover is, no. So, you will choose yes, no, yes, no, yes, no, yes, no, no, no, yes, yes, no whatever; what does it mean?

(Refer Slide Time: 16:24)



Suppose, your random numbers are such that y, no, no, y, y, no; that means, you will have a crossover here. So, the new thing will be 1 0 1 0 1 1; is it clear?

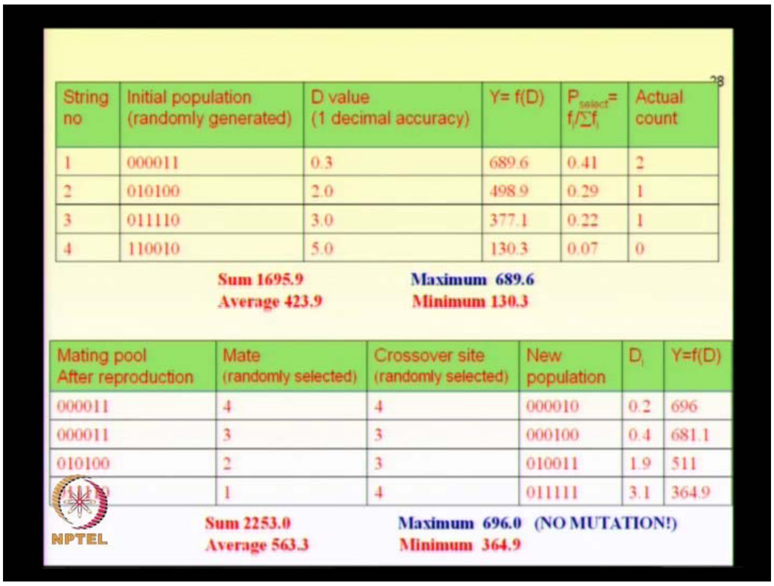
Student: Voice not clear.

Why? Ok.

So, this fellow becomes like this. How many of you did not understand? Good, so everybody has understood. So, this crossover also you can use, so I have used a simple, simplest possible crossover, so that it becomes easy for you to understand, but this is a correct way of doing it. The uniform crossover is actually used in programs, you can do it in the exam also. But, if you are really getting tensed up and you are not having time, just use a single point crossover. But, single point crossover, sometimes it will not give you the desired effect, the improvement in objective function will not be much.

Now, you have generated the new population. Just like you exchange the bits here, you exchange the bits between 3 and 2, you need not use the same crossover site; I used a different crossover site, get the new population.

(Refer Slide Time: 17:40)



String no	Initial population (randomly generated)	D value (1 decimal accuracy)	Y= f(D)	P_{select}= $f_i/\sum f_i$	Actual count
1	000011	0.3	689.6	0.41	2
2	010100	2.0	498.9	0.29	1
3	011110	3.0	377.1	0.22	1
4	110010	5.0	130.3	0.07	0

Sum 1695.9 Maximum 689.6
Average 423.9 Minimum 130.3

Mating pool After reproduction	Mate (randomly selected)	Crossover site (randomly selected)	New population	D_i	Y=f(D)
000011	4	4	000010	0.2	696
000011	3	3	000100	0.4	681.1
010100	2	3	010011	1.9	511
[illegible]	1	4	011111	3.1	364.9

Sum 2253.0 Maximum 696.0 (NO MUTATION!)
Average 563.3 Minimum 364.9

And, now, what you do is; so, this is the original set. The sum of Y was 1696, the average was 423, maximum was 689 and minimum was 130, fine.

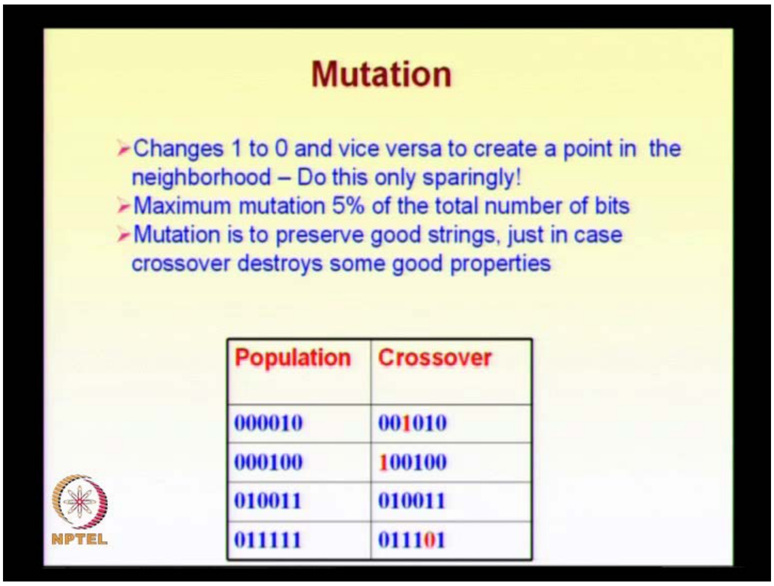
Now, let us look at the key statistics. After reproduction, the new population is like this. You can convert the binary, back to decimal, and then you can calculate the value of; you want to come here? You can get the value of Y. Now, if you see, the maximum did not improve much; from 689, it went up to 696, very marginal; but, the minimum has substantially increased, which means the average, the average is also going up. With just 1 iteration, you are able to get good successive GA, it is possible; this is the basic GA; has everybody understood?

Then finally, you can report the answer as, you can take the average of all these Ds and average of Y, and say, that is average, that is the average fitness, and the average value of D after 1 iteration. If you do 10 iterations, it will, it will remarkably improve; then, you may say, sir, for these simple such type of problems, I can use exhaustive search. It will be remarkably superior to exhaustive search, because as the iterations proceed, it will narrow down the solution, number 1.

Number 2, if you are using a dichotomous search or something, what will happen is, essentially the function has to be unimodal, otherwise you cannot apply, and your golden section search will come, because you do not expect the function to be zigzag. How are you able to confidently say, region to the left of this is eliminated, region to the right of this is, who told you that? Because, it is unimodal.

But, most real life problems are multi model. Genetic algorithms will not get choked, it will not, it will not struggle, it will not get throttles, it will not get scared, because of multi model this thing. You just keep evaluating and this, and then you are improving the solution. The key point is, you are not taking one solution at a time, you are taking several solutions. And, since, stochastic rules are applied, there is a good chance for getting convergence.

(Refer Slide Time: 19:44)



Now, sometimes I told you, already you must have, already at the end of first or second generation itself, you must have hit upon the answer, but since every string is subjected to change, the good answer will, the good solution will also get broken down, isn't it? You do not know when you have reached. So, actually, in some iteration, there may be a degeneration of values, locally there may be; generally, there will be an improvement. Sometimes, if you already reached the best, that fellow cannot be preserved. I told you there is something called elitist strategy to preserve, right; otherwise, what you do is, sparingly use your, you use mutation.

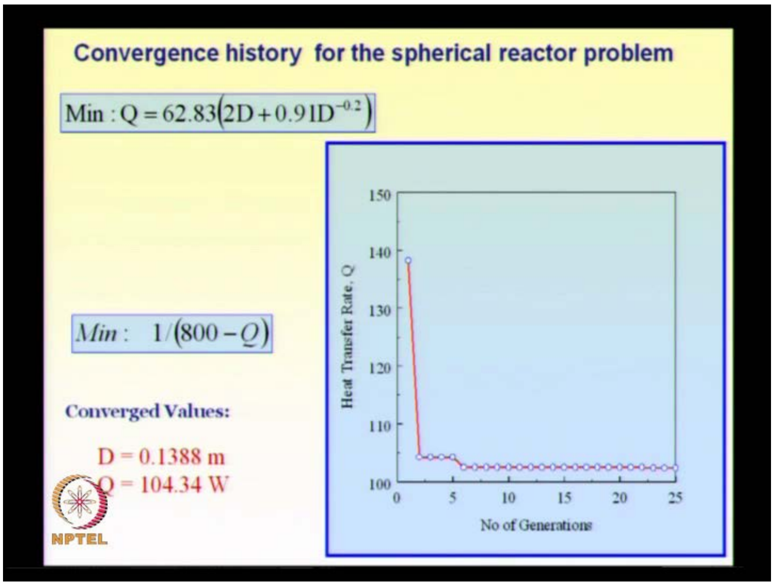
What is mutation? 0 is converted to 1 and 1 is converted to 0, maximum of 5 percent is allowed; maximum 5 percent of the bits, you can do mutation. So, how do you implement it in a, using a random number table?

Student: Choose the location and so on, the decimal value into number of, total number of bits.

Decimal value into total number of bits; and, what is the probability you want? If the probability is less than point, if probability is less than 1 by 20, if the probability is less than 0.05, if the random number is less than 0.05, change, else do not change; in 20 times, it will come only once. If it is a really random number table; it may come once or twice, that is, ok; are you getting the point?

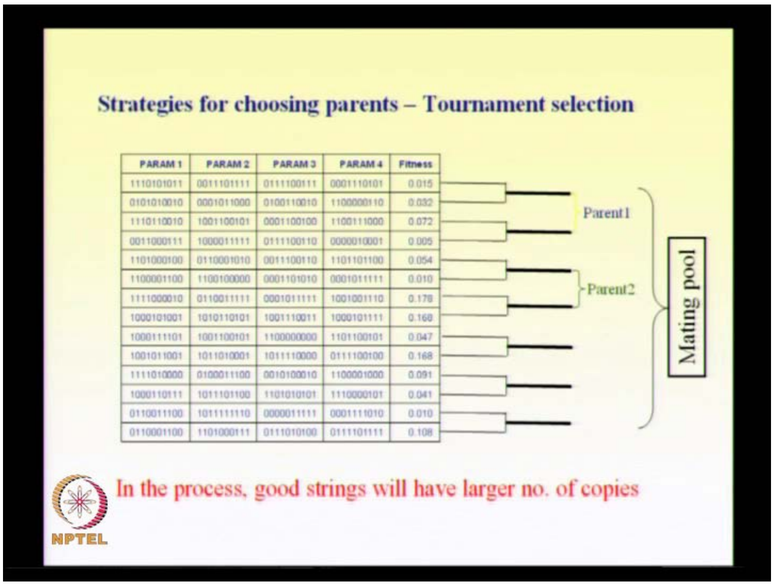
Now, you know how to combine probability with a random number, right. So, if you are fixing the mutation at 5 percent, then you should look at the random number 1 by 20 of 1, that is 0.05; so, you declare that, after you have exhausted 30 random numbers, you will start looking at the 31st random number, so you have finally, the final population 24 bits, 31 st random number; random number, less than 0.05, no mutation; random number 32, less than 0.05, no mutation. Whenever it is less than 0.05, yes; mutation, 0 is converted to 1. Do not apply in the incorrect way, then 95 percent of the bits you will change; so, once the algorithm starts working, you will figure out.

(Refer Slide Time: 21:59)



So, this is what we have done, this is what we have done; now there will be lot of fun, we will solve a problem from, on our own; we will construct the, we will construct whatever I shown you here, for a simple problem. So, this is what we have done, it converges after 5 or 6 generations, we also solved it as a two variable problem.

(Refer Slide Time: 22:18)



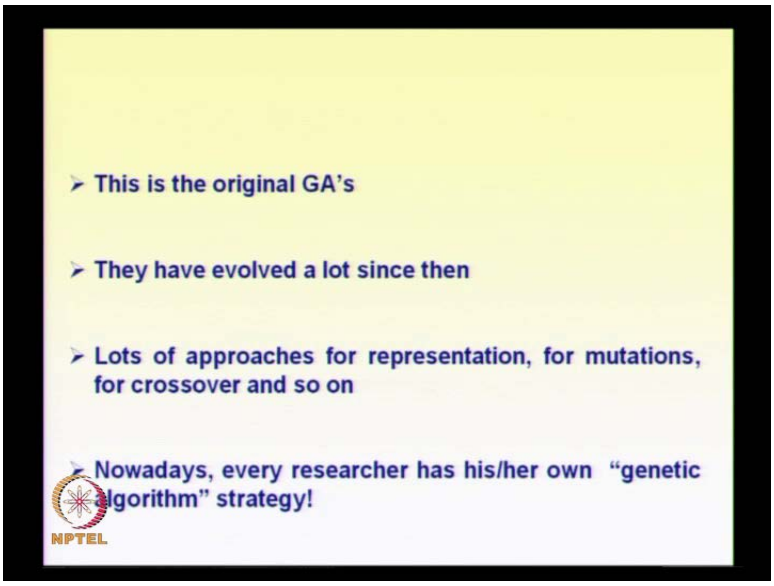
There are some strategies for choosing parents, so this was very clear, because we had only 1 variable, but suppose you have more variables, this was a radiation problem which we solved. One of my PHD Student's Swaminathan, he solved it some 5 years back. So, if there are 4 parameters, we do what is called tournament selection.

If there are 14 pairs, now if there are 14 pairs like this, you want to choose the best, what you do is it just like t 20 you have a knockout. Two players at a time, two players at a time they will play, whoever has a higher fitness will go to the next one. Whoever has a higher fitness go to the next one. Then you will eliminate 7, from 7 again you take 2, 2 at a time; are you getting? This is like a knock out.

For a single variable, single variable problem it was very clear; how to choose the mates and all, how to choose the next population mating and all that, but it is not so easy. So, lot of strategies have been developed to pick the better ones in the population, the tournament strategy, the tournament strategy is very popular in GA. This helps you to get the mating pool,; are you are you getting the point?

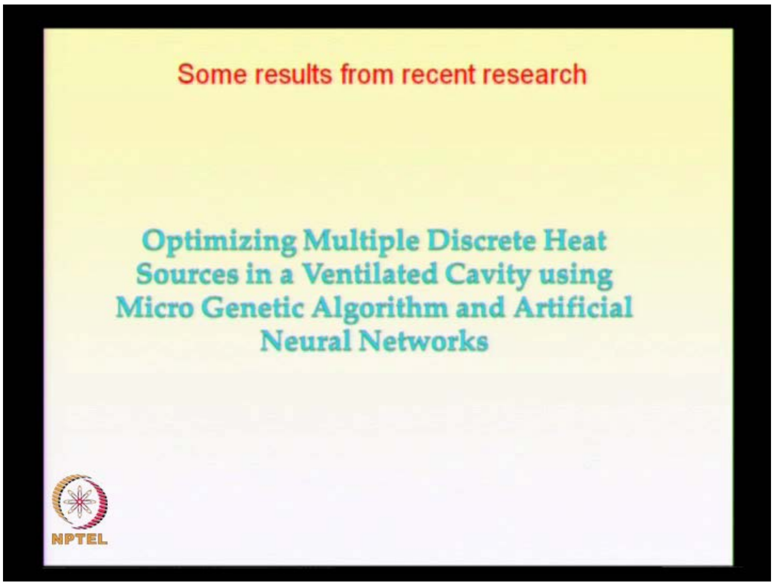
See, for example, if you have 40 variables or 50 variables, and you want to use 2 n to 4 n number of solutions, you will use 200 solutions, you will get 200 solutions each time. You have to write a program to put them in descending or ascending order of fitness, fine; and then, you do not basically choose all the top 20; sometimes the bottom most 20 will also have some desirable characteristics. If the chromosome is long enough, there may be some 0 1 0 0 in the middle, which may be good; you do not want to lose that, that is the beauty of GA. We do not want to outlay reject others, first generation itself will say these 4 are the best, this is the solution, that is not what we want to do; are you getting the point? So, diversity is very important. Therefore, how to get the mating pool? People have developed strategies; how to, how to develop crossover? People have developed strategies. So, there are so many techniques for developing, for mutation, crossover and so on.

(Refer Slide Time: 24:38)



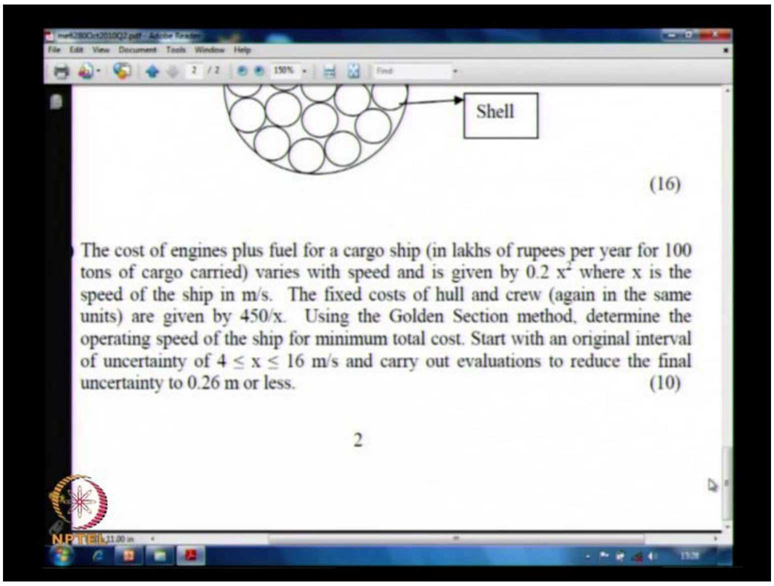
So, what I have shown you is the original GA. They have evolved a lot since then. Nowadays every researcher has his own GA. So, all these I have said, I have said.

(Refer Slide Time: 24:44)



I have some results to show; it is basically, I have already put this on moodle. These are some of the research results which we obtained using GA. But, right now we will solve the problem.

(Refer Slide Time: 25:05)



This is the second quiz, we will take one problem, which you solved using the golden section search. We will try to solve the same problem. Please take down this problem, the cost of engines plus; simple problem, so you know the answer also. The cost of engines plus fuel for a cargo ship, so please copy the first four lines. All of you have the random number table, yeah; I need one; you all have, Deepak, Vikram, Abishek; yeah, you can keep one more. So, how long is your review Ashutosh, how long is your review?

Student: Sir, it will be around, I do not know that depends.

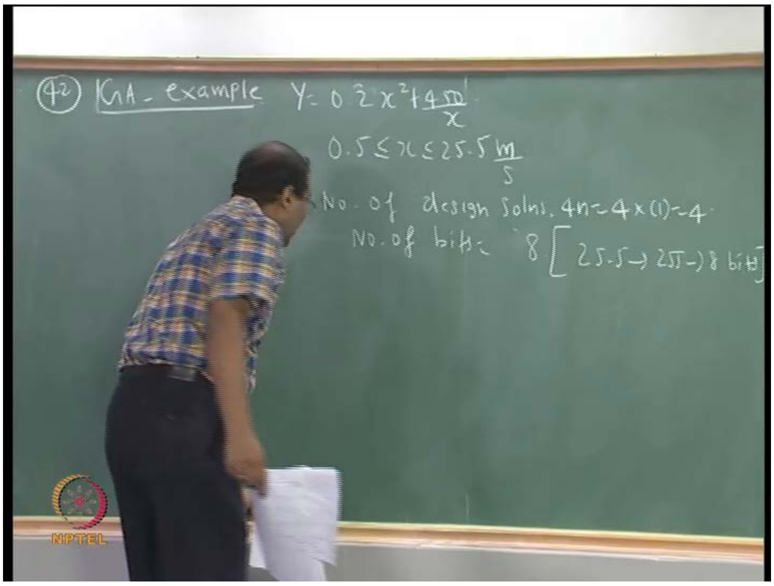
Then, you can comeback at 2:30 if you want, no.

Student: I do not know whether it will take half an hour or more, it is really a very lengthy method.

So, the cost of engine plus fuel for a cargo ship is like this. The fixed cost of hull and crew are given by 450 by x. Using genetic algorithms; so, the question continues; after 450 by x, after 450 by x full stop. Using GA perform two iterations, using GA perform 2 iterations to determine the optimal operating speed of the ship, using GA perform 2 iterations to determine the optimal operating speed of the ship. What is the problem number?

Student: 42.

(Refer Slide Time: 27:21)



Perform 2 iterations to determine the optimal operating speed of the ship. The original interval of uncertainty may be taken to be, 0.5 less than equal to x less than equal to; the original interval of uncertainty can be taken to 0.5 less than equal to x less than equal to

25.5 meters per second. 1 decimal accuracy is required for representing x, 1 decimal accuracy is required for representing x.

We can take this off, I will use the; you need not; I will use it again, let it be on; projector, you can turn it off, or you can put it in sleep mode. I will keep this like this; we can have the lights.

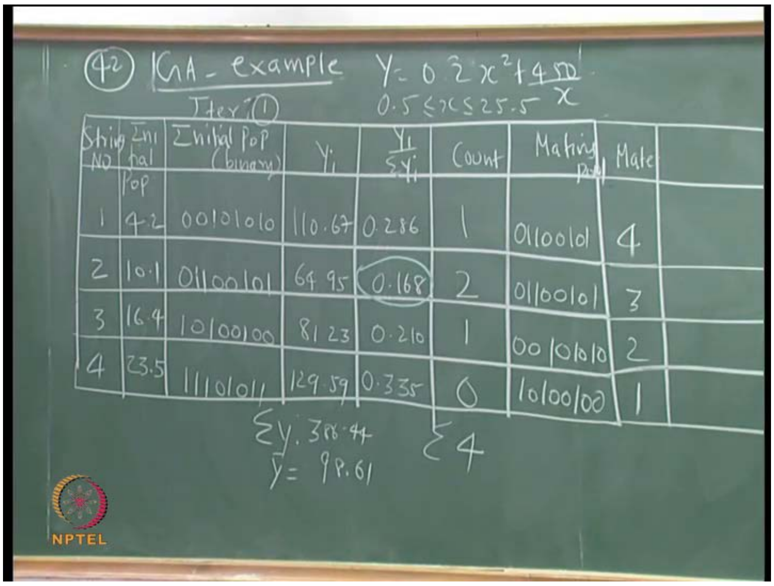
What is the highest value of x? 25.5; 1 decimal accuracy is required, what is a number we are looking at? 255; so, what is the largest number with 8 bits? 255, so how many bits are required? 8. So, GA works with 2 n to 4 n solutions, we will work with 4 n, n is the number of variables, so number of solutions, number of designs, I am writing in short hand notation, 25.5.

Student: Voice not clear.

N is the number of variables, agreed?

Now, you have to start the GA. I will erase this, let us keep this.

(Refer Slide Time: 30:15)



So, string number; I will give you the initial population, so that everybody, who is along with me, right. 1, 2, I am using 4 strings. I want it to be, so 0. Either you can use the random numbers and generate the 8 bits, or you can equally distribute the 4 solutions in interval, 0.5 to 25.5. So, I have taken 4.2, 10.1, 16.4, 23.5. Here, we are allowed to, we have 8 bits, so we can do 1 decimal.

Student: Suppose you have 2 variables in the 8 bits, does it mean that 8 is of one variable or 8 is for another variable?

No, no, no, no. You can put that string side by side

Student: So, it is 4.

No, 8 number of solutions, that is a population size, but the strings will be written x 1, x 2. I gave you the binary representation, right.

Student: 8 for x 1, 8 for x 2.

8 for x 1, 8 for x 2; 8 for x 1, 8 for x 2, 16 bits, but 8 solutions, so it will become very messy, do not worry, I will not ask 2 variable problem in exam, you want? But, that is a way to do it, what you are saying is correct.

Student: Why do we need more solutions?

That is what, that is what Goldberg and Holland, 2 n to 4 n. If you take 4 n, it is it will converge faster and diversity will be more. 2 n will be like almost like; 4 n is, 4 n is good, though some optimization may be possible.

Now, can you convert it into this? So, when I put; see, 25.5, I am representing as 8 bits, means, 8 bits is 1 1 1 1 1 1 1 1, therefore when I represent 8 bits, I will convert the; watch carefully; when I have 8 bits, I will convert the binary to decimal and divide by 10, that you must have in your mind. So, what will be this? So, you have to get the binary equivalent of 42, what is it? 0 0 1; everybody is through with this? So, 0, 2, 8, 10 plus 32, 42; 42 divided by 10, 4.0, is that ok, fine.

This is 0 1 1; 1 0 1; please tell me if I made a mistake. Is it ok, have you checked all the 4? Now, please calculate the value of Y, for each of this. Y is 0. 2 x square plus 450 by x; 110.67; is it correct, did I make a mistake, fine; Robin, is it ok?

Student: How to use random number table and generate bit by bit?

Yeah, random number table, you do it bit by bit, know; if it is, see there are only 2 possibilities 0 and 1, right. Robin has a question, sir how to use the random number table and generate this bits; there are only 2 possibilities 0 and 1. So, the probability has to be 1 by 2.5. So, you set the probability as, probability greater than point 5, the bit is 1; the probability is less than point 5, the bit is 0. Start from left to right; that is from top to bottom, here left to right. So, you will get 0 0 0 1 0 1 0 0, first 8, over.

So, the next, the next string will be 9th one, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0 0 0 1 1 0 1 1, correct, that is a way; but, do not break the sequence, we have to go in that order; by the time you do mutation, crossover, if it is exhausted, hopefully it does not exhausted, you have to come back to the, it would not get exhausted.

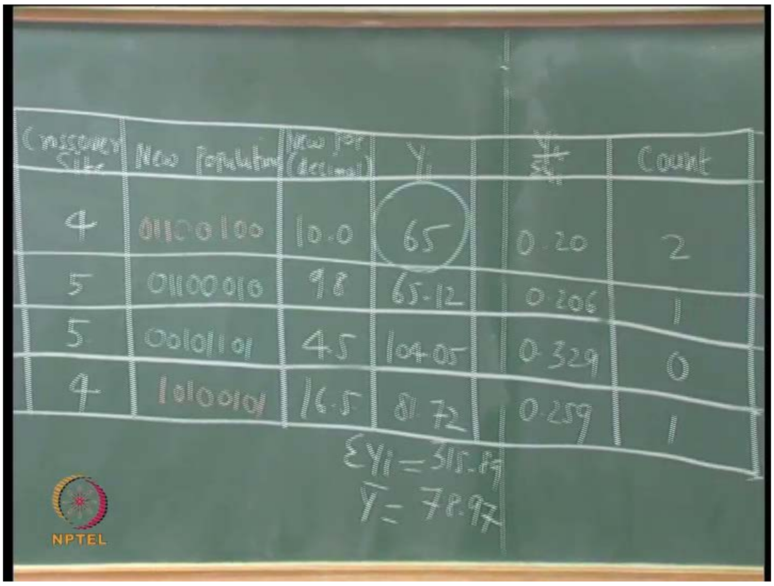
Now, how do you check whether I do not have any bias, check the number of 1s and the number of 0s, how many 1s are there? 16, how many 0s? Very good, so I am not done any hanky panky, it is fine. If the number of 0s equal to number of 1s, that means, essentially what the random number table is capable of doing, I am, I have done; is it clear, Robin.

Values of Y; are you alright? Can we get sigma y, 386.44, y bar 98.61, please keep this. What is the average fitness? Please note that the original GA is for maximization, we are trying to do minimization. Now, you can do y divided by sigma y, 0.286, correct; so, column number 5 is called the relative fitness, relative fitness; the sum of all the relative fitness will be equal to 1; who is the fittest now, for our problem? Yeah, please do not be mislead, please look at whether it is a maximization or minimization problem. For a minimization problem, this fellow is the fittest fellow. In fact, it is very close to the correct answer of 10.4 meter per second.

But, we are feigning, we are assuming that we do not know the answer, we have to proceed. Here, it is so obvious, because that we can do it by calculus. Now, I want to have a count, what is this count? I want to have a count, so that I decide on what will be the mating pool. So, this fellow is very strong, so we will make twice, he will come twice. This fellow will come once, this fellow will come once, so this sigma must be 4 always, this sigma must be 4 always, constant number of, constant number of members in a particular generation.

What will be the mating pool? So, 0 1 1 0 0 1 0 1; I put exactly in the 2 1 1 order, I am changing the order now; I am changing the order, in the order of decreasing fitness. Now, this is still iteration 1. In the exam, you can use the paper this way, it will be easier; in the exam you can do it this way.

(Refer Slide Time: 41:06)



So, what do we do now? Mate. So, I am putting 4 for this, therefore 1 for this, 3 for this and 2 for this. For the first guy, the mate has to be 3 or 4, because 1 and 2 are the same guy. Once you have put 4 for 1, for the 4th guy, it becomes 1, and second and third it becomes automatically. No crossover, I am using only single point, so I am having crossover here 4 4, I am having here 5 5; that is, at the end of the 4th bit, that is the cross, so I will say crossover site.

Why do you say crossover site? I will say crossover site, only if I use single point crossover, but you are encouraged to do uniform crossover, bit by bit in the exam. If your funda are strong, you can do that; single point is adequate. I will not penalize you, if you use single point. Now, we will use different color chalk. 4th, so, who are the new guys? So, I will say new population, so new population is; yeah, can I complete. 0 1 1 0 0 1 0 0. What about this? 1 0 1 0 0 1 0 1. So, this random crossover is site number 3, I mean at the end of the 3rd bit,

Student: 5.

5, ok. 1 2 3 4 5, fine. Yes, I am asking a 20 mark question in the exam, and straightaway tell you 1 variable, you, everybody you can score 20. If you can do it now, that is it, linear programming, dynamic programming, genetic algorithm, 10, 10, 20- 40 marks already, simple. Say, you should know how to use the random number table; even if you do not know, it is fine, you can use this simple single point crossover and; yes, Abhishek any problem?

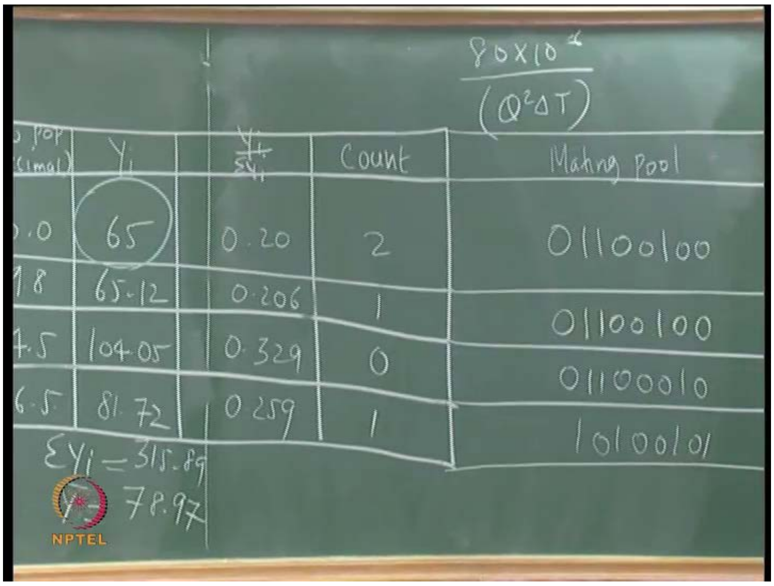
Now, new population decimal; yeah, please convert it to decimal, 10.0. Now, what is y i? 65; correct? Please check. What do you have to check? Somebody should check the 4th value, somebody else should check the 2nd value. Chaitanya, is it ok? All of you got this, 4 values. Now, do the p select; relative fitness; what is sigma y?

Student: 315.89.

Vikram you did not get, which one? How many of you are still doing? Yeah, please complete it; sigma y is 315.89, y bar is, Student:78.97. Yes, the beauty of this is, in just 1 iteration, the sigma has come down from 386 to 315, the average fitness has come down from 98 to 78.97. I may sound controversial; we want the average fitness to go down, because we want we are looking at a minimization problem. So, it has come down from 98 to 78, and the sigma has also come down.

The most important thing in genetic algorithms is, the variance in y or the difference in y between the minimum maximum will drastically come down. It was like 64 to 129, what was the difference? 65; what is the difference here? 40, so, it will come down further. In fact, you are supposed to do 1 more iteration; do not worry about the time. Now, you understand how the basic GA works? Alright.

(Refer Slide Time: 47:43)



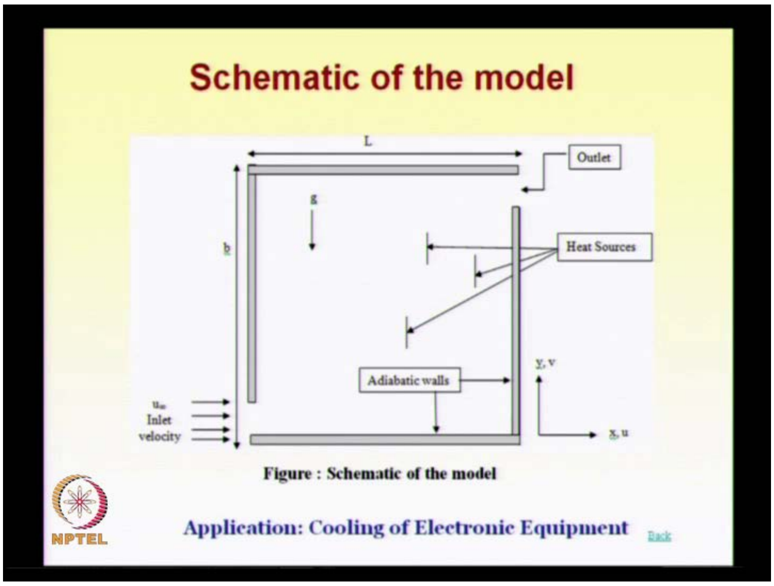
Now, if you do the count, so the mating pool will become; now, you can have a different, you can have a double crossover, you cannot crossover at the end of 2nd and 5th bit, 3rd and 6th bit, or you can have a single point crossover. You can decide your strategy, and you write the strategy; even in the exam you can say what the strategy you are using. Second iteration, it will remarkably come down, the value of y. In 3 or 4 iterations, you will get an average fitness, which is very close to the correct answer. You may, I come again, you tell me, sir I can do exhaustive search, whatever, but for a multi model problem, it will not work, the other, your traditional algorithms will not work.

The beauty of the GA is, it searches in the whole solution space; and then, even if some solution is very weak, it is not straightaway discarded. How it is not straightaway discarded? It can come, it can come in the form of mutation, or some good strings, some good strings, substrings of a string; that is, if 0 0 0 1 0 out of 12 bits, if there is a particular 0 1 0 pattern, it can be picked up and so on; that is why we rank them, the tournament, and all that.

In the tournament strategy, in the tournament strategy, it is not, the tournament strategy will result in good mating pool, compared to just putting them in descending order; are you getting the point? Because, between every 2, 1 is eliminated. We do not want premature convergence, at the time, at the same time, we want to have good convergence, I mean these are a conflicting criteria, but we are able to have best of both of else in GA; is it ok.

Now, all of you have got an idea of, how the algorithm works. I have the actual solution to the second iteration; you can do it, you can do it as an exercise; you can complete it. And, I will, in the next class, first 5 minutes, I will show you some results which we got, and then there will be a 5 minute break, you can come back at 2 o clock.

(Refer Slide Time: 50:39)

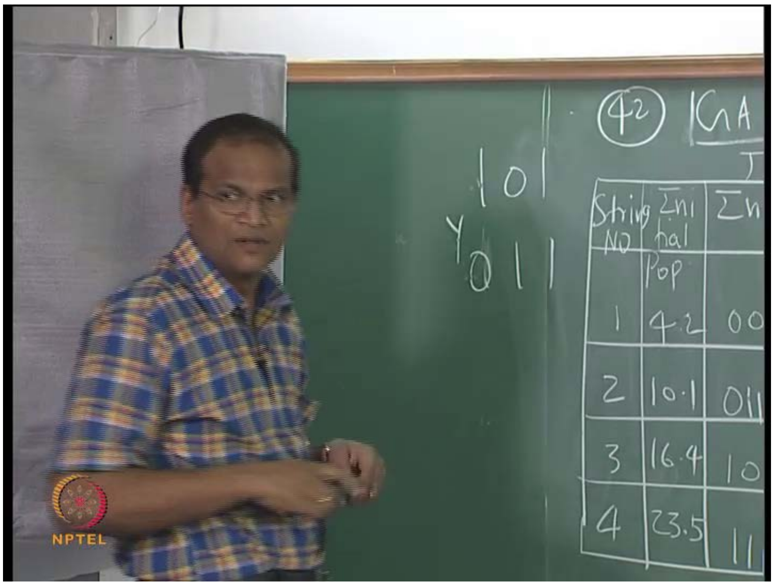


This was the; if you have any doubts, you can ask me now, Senthil.

Student: If you are doing the cross over at every point, then so the random number should be less than 0.5 from the first to the second, is it greater than point 5 and less in the second to the first.

No, if you want to exchange or not, is divided by yes or no criterion, so either you decide less than 0.5 greater than 0.5; if it is yes, from the first thing, the first bit of the second, the second one will go the first, what is the problem?

(Refer Slide Time: 51:15)



Suppose, you have a S, suppose you have a S, this 1 will become 0, and this 0 will become 1; it is applicable for both the, both the strings.

Student: No, sir both will be now

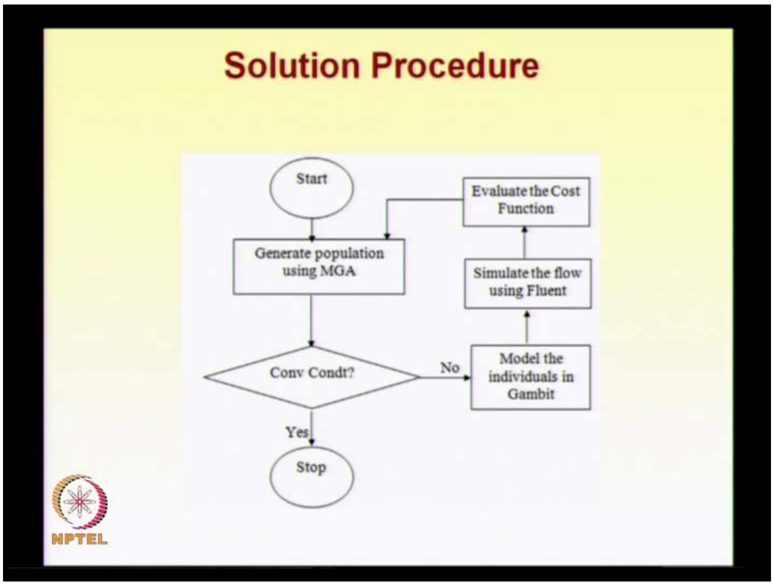
Both will be interchanged.

Student: It depends on the under and over.

So, this is basically, if you consider your desk, you will come back and; just wait, you can sit down somewhere, Abishek.

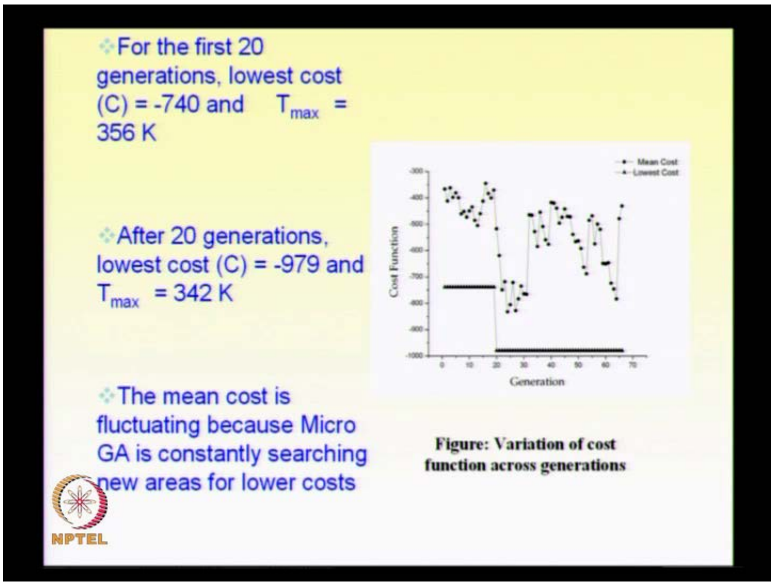
See, suppose you have got assorted electronic equipment in the, in the cabinet of your desktop or whatever, if they, and suppose each of which is generating some heat, you want to look at the optimum position of this, such that maximum cooling is achieved. When is maximum cooling achieved? When the temperature is lowest, that is when nusselt temperature is highest. But, you can see, there are infinite possibilities for these 3 things to be kept in this space. Now, we are trying to see whether it can be, this problem can be tackled using genetic algorithms.

(Refer Slide Time: 52:53)



So, there is an airflow here. It is a very simplified configuration of an actual electronic equipment. There are only 3 heat generating chips. We want, we want to minimize the maximum temperature, the minimax criterion. So, we solve this problem using fluent, and then we got the temperature distribution around each of these chips, heat sources. And then, so what we do is basically take some arbitrary positions, run the fluent and get the maximum temperatures. Then, run it using the GA, generate 3 new positions. And then, take it to fluent, find out the maximum temperature, change the position, change the position, change the position.

(Refer Slide Time: 53:14)



And, this is how we got. The cost function is basically the, cost function is basically the minimum temperature, so minimizing the temperature, so we, so we got something like this. But, what happens is, even after 30, 40 generations, it oscillates. Therefore, it is extremely time consuming, because we do not, we cannot, we cannot afford to run 100s of generations, because everything is a fluent solution.

So, what we did was, we took some 50 or 100 combinations of positions, we first ran the fluent for each of the, for all these 100 cases, and we developed an artificial neural network model for this. What does the artificial neural network model give? You give the, you give the 6 coordinates x 1 x 1, y 1 y 2, x 3 y 3 of each of these chips, it will directly give you the maximum temperature. Now, you power the GA with this, with the output of the artificial neural network. So, you combine the neural network with the genetic algorithm, so the genetic algorithm is driven, the genetic algorithm is driven by the neural network, rather than by the fluent; are you getting the point?

(Refer Slide Time: 54:14)

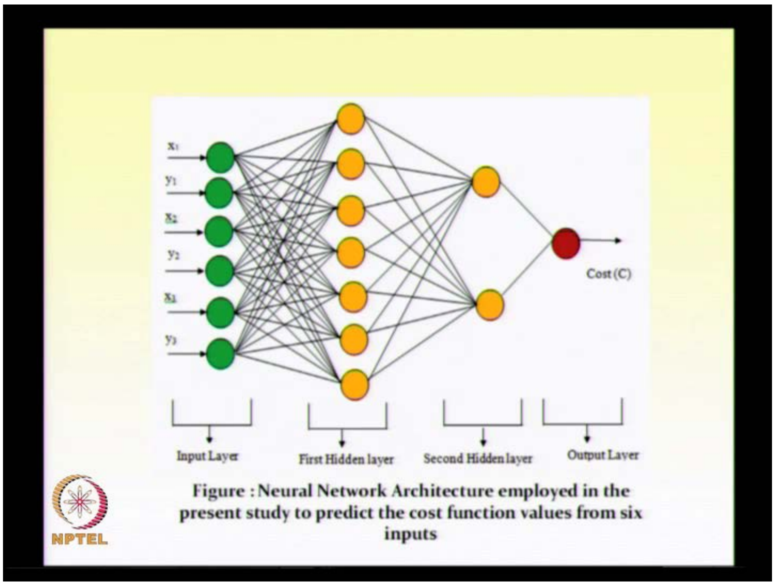
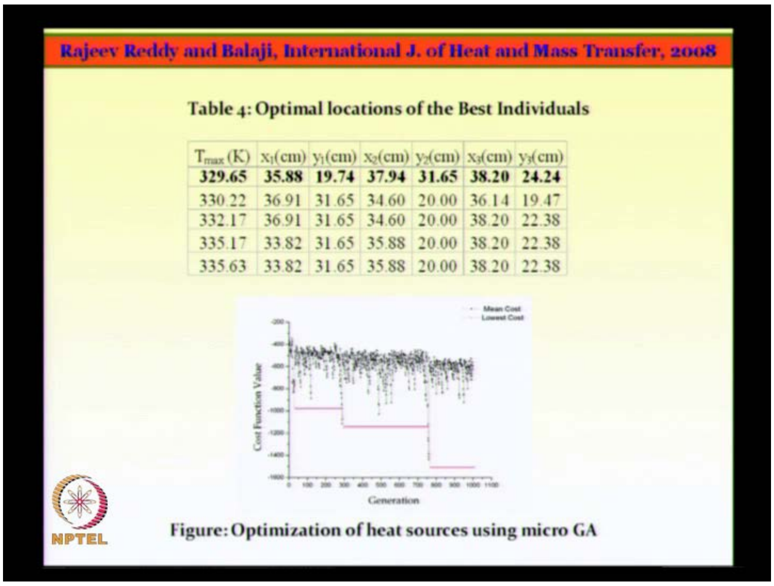


Figure : Neural Network Architecture employed in the present study to predict the cost function values from six inputs

So, you have a neural network, you give the 6 positions, from the fluent solution, it will give you the maximum temperature.

(Refer Slide Time: 54:22)

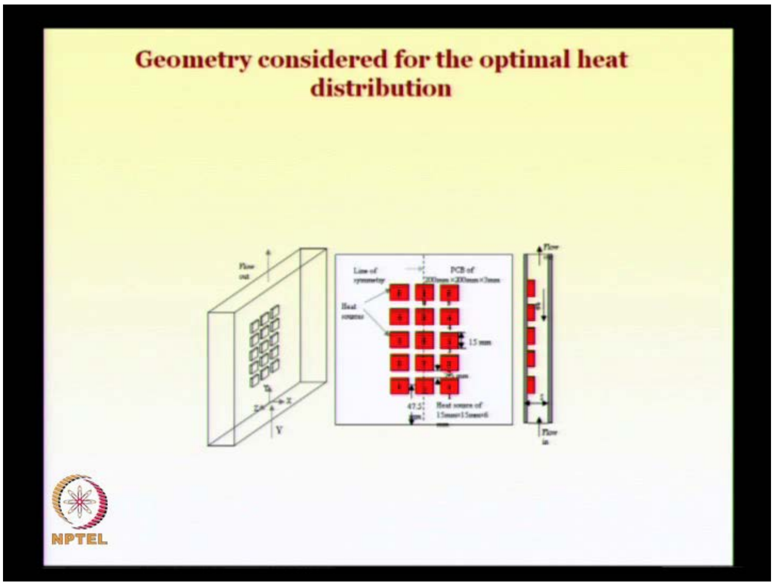


So, the optimal location, this is international journal of heat and mass transfer, you can; he was a B Tech student, 2008 batch; he also got the C. S. Krishnamurthy award for the best project in genetic algorithms for that year. So, you can see that, with 1000, I can generate 100, I can run it for 1000s of generations, because it is very fast; now, I have a neural network, right.

(Refer Slide Time: 54:50)

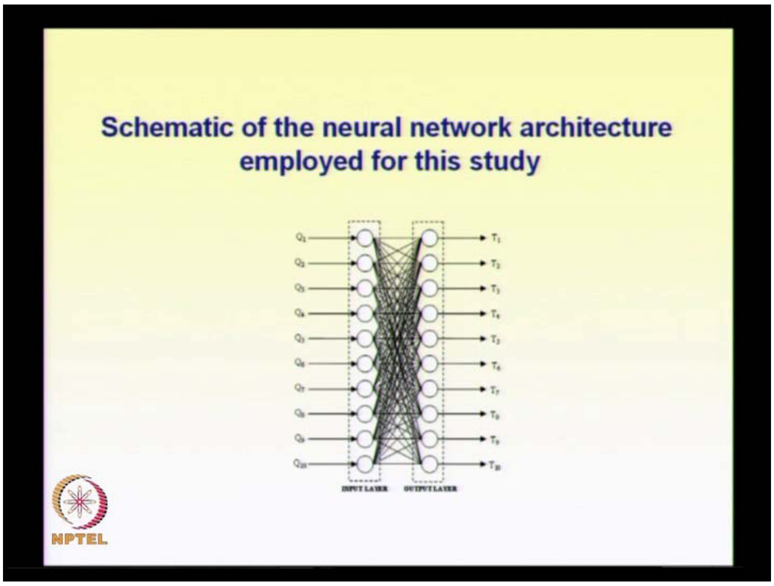
So, this was, this material is based on the PhD work of my former student T.V.V. Sudhakar, who also did experiments.

(Refer Slide Time: 55:00)



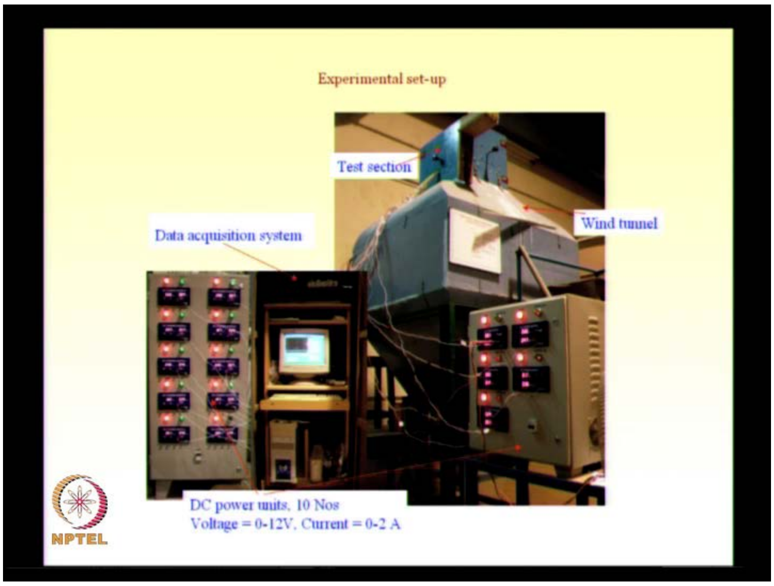
So, basically he took a configuration like this. So, there is a, there is a plate, a substrate, a substrate like this, on which heat generating components are there. There are 15 of them. If you want to maximize the total heat dissipation rate from each of these components, such that the maximum temperature is less than 85 degree centigrade on any of these chips, how will you know about? Or, if you have a total of 15 watts to be dissipated, how will these 15 watts to be dissipated among these chips? These are the questions we tried to answer.

(Refer Slide Time: 55:46)



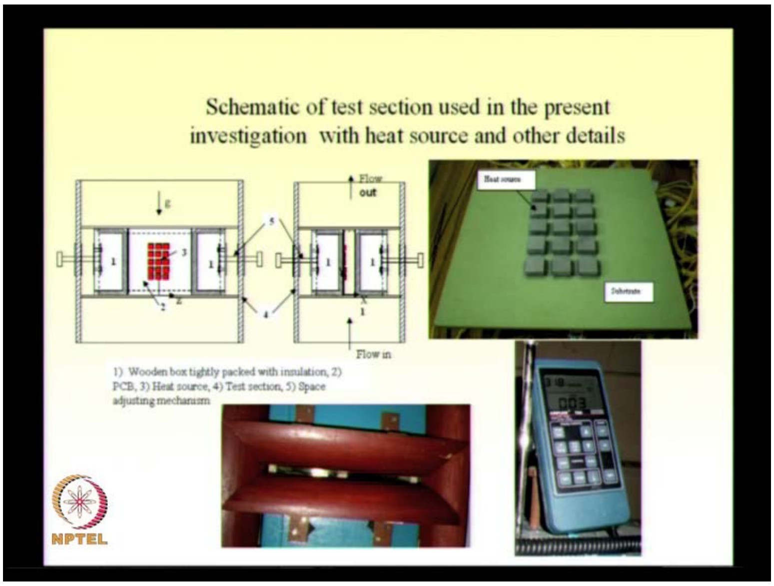
So, again, governing equations, we solve them using fluent, gradient dependence and machine dependence studies, then benchmarking with standard results. Then, because there are far, too many combinations, again we use a neural network, so neural network combined with; so, you give Q 1 to Q 10; there is a symmetry, so center, right side and left side are the same. So, if you give 10 inputs, it will automatically give you the output of the temperature. This is obtained from fluent. And now, you run a neural network model for this, then neural network model replaces the fluent, that is called a fast forward model, it is called a fast forward model.

(Refer Slide Time: 56:19)

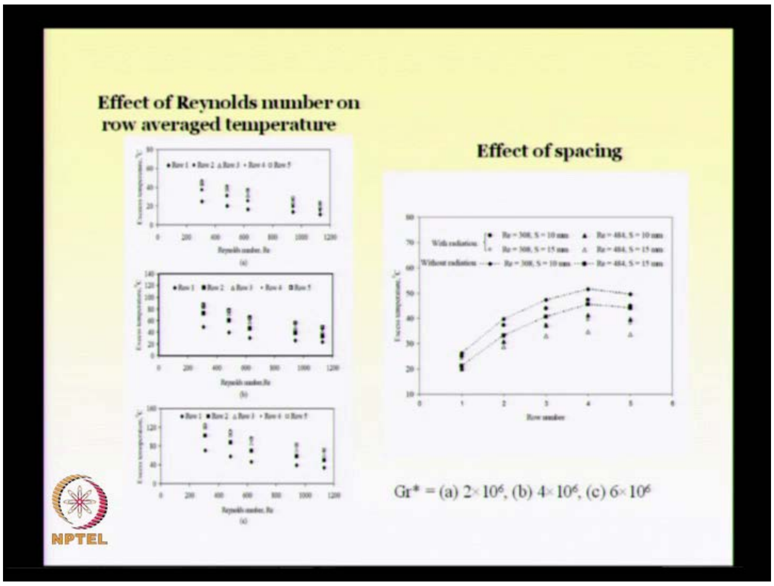


Then, the performance metric, mean relative error, and variance and all this. And then, how do we decide the optimum weights in the neural network? Performance of the network, we also did, it was not just theoretical prediction; this is the wind tunnel, which is, there in a heat transfer lab, it is a vertical wind tunnel, we can do both natural convection and mixed convection heat transfer studies; these are the DC power supplies.

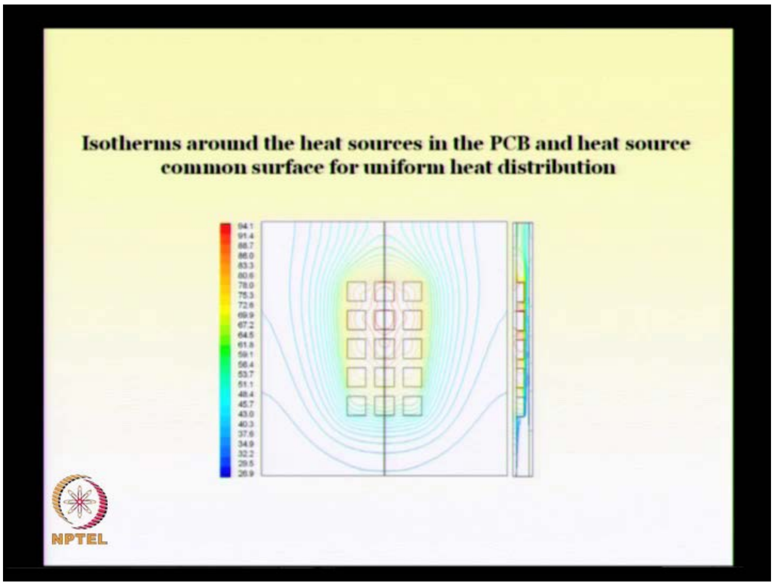
(Refer Slide Time: 56:32)



(Refer Slide Time: 56:44)

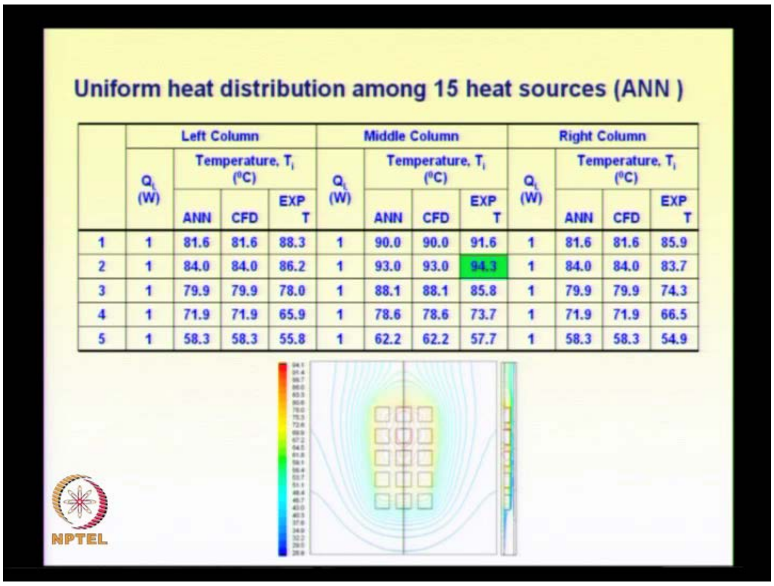


(Refer Slide Time: 56:50)



So, this is your, this is the setup, so this is the bell mouth, these are two views of the setup, these are the heat sources in the substrate. So, we generated the, whatever optimum we obtained, we also put it on the wind tunnel, and we got the measurements, and the theory and measurements agreed very well. These are some stream line and isotherm plots to GA; so, we combine GA with neural network; this is the algorithm.

(Refer Slide Time: 57:07)



	Left Column				Middle Column				Right Column			
	Q_i (W)	Temperature, T_i (°C)			Q_i (W)	Temperature, T_i (°C)			Q_i (W)	Temperature, T_i (°C)		
		ANN	CFD	EXP T		ANN	CFD	EXP T		ANN	CFD	EXP T
1	1	81.6	81.6	88.3	1	90.0	90.0	91.6	1	81.6	81.6	85.9
2	1	84.0	84.0	86.2	1	93.0	93.0	94.3	1	84.0	84.0	83.7
3	1	79.9	79.9	78.0	1	88.1	88.1	85.8	1	79.9	79.9	74.3
4	1	71.9	71.9	65.9	1	78.6	78.6	73.7	1	71.9	71.9	66.5
5	1	58.3	58.3	55.8	1	62.2	62.2	57.7	1	58.3	58.3	54.9

(Refer Slide Time: 57:15)

Optimum heat distribution among 15 heat sources (ANN plus GA)

Left Column					Middle Column					Right Column				
Q_i, W		Temperature, T_i (°C)			Q_i, W		Temperature, T_i (°C)			Q_i, W		Temperature, T_i (°C)		
Num	Expt	ANN	CFD	EXPT	Num	Expt	ANN	CFD	EXPT	Num	Expt	ANN	CFD	EXPT
1.0948	1.107	77.3	77.1	84.0	0.6659	0.640	77.0	77.7	78.3	1.0948	1.089	77.3	77.1	82.7
0.5879	0.592	73.5	73.3	75.1	0.5843	0.594	77.2	77.0	76.8	0.5879	0.584	73.5	73.3	72.9
1.0272	1.021	77.3	77.1	78.0	0.0229	0.024	70.4	70.2	64.2	1.0272	1.081	77.3	77.1	74.9
1.0342	1.035	77.1	76.7	72.7	0.6274	0.606	77.0	77.0	69.8	1.0342	1.010	77.1	76.7	70.5
[illegible]	2.015	77.0	76.0	76.3	1.6177	1.586	77.4	76.4	71.1	1.9968	2.054	77.0	76.0	75.9

And then, the most important thing is, so if you have uniform heat distribution, you just give 1 watt to each of this, the total is 15 watts, the maximum temperature is 94, it is too much; but now, you can reduce it to, you can reduce it to not only, not only 84, but now you can see that some places you have got less than 1 watt, some places you more than 1 watt; but this you cannot have a Desi method of, you know that wherever coldest air meets the chips, there you can load it more. As the air picks up the heat, it becomes hotter, its capacity to cool the other chips reduces, but this is common sense.

But, when you have so many heat sources, your common sense alone will not guide you to the correct solution, tools are available, make use of the tools; use GA, have fun.