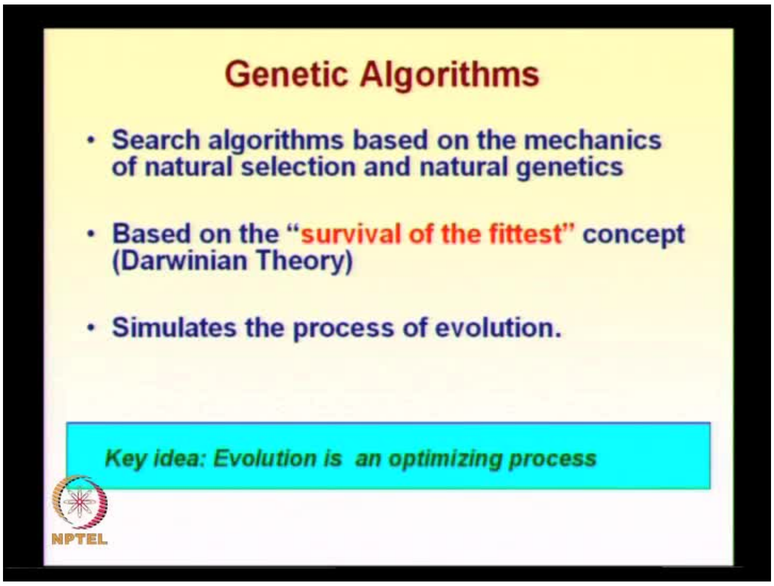
Design and Optimization of Energy Systems
Prof. C. Balaji
Department of Mechanical Engineering
Indian Institute of Technology, Madras

Lecture No. # 38
Genetic Algorithms

Today, we will look at a nontraditional optimization technique. A very powerful nontraditional optimization technique is a class of algorithms, which are based on the, which mimic the process of evolution, right. So, you, you look at biological system and find out the features of biological system, and see whether these can be implemented in engineering. So, anything which looks at the process of evolution and we try to develop algorithms or we try to do computing based on that, these are known as evolutionary techniques.

So, evolutionary optimization technique, there are several of them; most important among them is the genetic algorithm, also called GA, right. So, it is relatively new, may be about 30 years old; figured out in 1975, but it is gaining popularity in the last 15 to 20 years.

(Refer Slide Time: 01:03)



So, what is G R? What are G S? Genetic Algorithms are basically search algorithms. So, they are not calculus based algorithms; we do not use derivative knowledge or hessian matrix, second derivative, third derivative and so on. It is essentially a searched, search technique. So, the logic is, any search technique should have a logic; the logic varied, right, previously, for two-dimensional search, you searched around a particular point, East, West, North, South, North East, North West, South East, South West. And then, you proceeded in one direction or that is hill climbing, or what did it was, you can consider two points and founded whether why increases or decreases, remove one portion of the interval.

If it is a two variable problem, one variable at a time you can do this. So, you can have an efficient single variable search technique, which could be broadly based on either hill climbing or region elimination method. So, the logic as for as genetic algorithm is concerned is, it is based on the mechanics of natural selection and natural genetics; so, it is, right.

So, it is based on the survival of the fittest concept, which is basically the Darwinian Theory. Only the fittest will survive and reproduce and procreate, and successive generations will become better and better compared to previous generations. This is true, this is true, if you look at any parameter; for example, the average life expectancy of an Indian male is now 68 years, it is only 40 years of the time of independent. So, the life expectancy in Japan is about 88 years; there are 100, 1000 people, more than 100 years living in Japan. And then, the average, the probability that child born in Germany, today, will live for 100 years is more than 0.5.

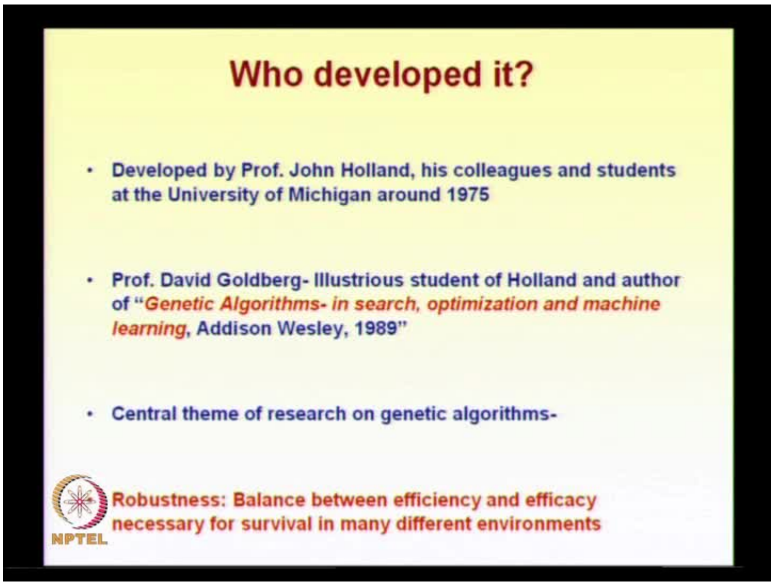
We cannot say that, all the diseases have been conquered; several diseases have been conquered, new diseases have come, but then medical research is going on; lot of people live with diseases, they do not drop dead like flies. So, people with pacemaker, this thing, well replacement, artificial thing, lot of god's original equipments is replaced, but still people live, right, and after by pass and so that, the quality of life is very good. So, because of all these; so, outside of Darwinian evolution, there is intervention by man also.

But, it cannot be denied that successive generations are become increasingly better. The mosquitoes are, of today, are much better than the mosquitoes of yesterday; in the sense, that they are able to cheat the all out, and; from our perspective, it is very bad, but for them, they all, they know that, all out is this thing. So, they have, Biochemical changes are taking place, because only those mosquitoes which survive, they will reproduce and all; then after some time, this, they do not care about this all out; they sit right on the all out, right, I think they do that. So, they are the having; so, that means, you have to change the chemical or you have to do some other this, right.

So, basically genetic algorithms is, it simulates the process of evolution; so, we simulate or mimic or imitate the process of evaluation, so the key idea is evolution is an optimizing process, right. If evolution is an optimizing process, and successive generations are becoming better and better, each generation is like an iteration in numerical methods, right. So, we have 5 generation, which equivalent 5 iterations in the numerical technique.

So, with each iteration, there is a progressive improvement of the objective function; that means, it is more or less like a hill climbing technique. You do not eliminate; it is an hill climbing technique, these successive generation are better and better; therefore, successive iterates of our design variables. So, if, Y is function of x1, x2, x3, x4, you start with some combinations of x1, x2, x3, x4, to get the value of Y. As you apply the genetic algorithm, its successive iterations, the Y will keep increasing, if it is a maximization problem. So, that is the key idea.

(Refer Slide Time: 05:14)



Who developed it? Developed by professor John Holland, his colleagues and students at the University of Michigan, and in 1975 – it is about 35 years old. So, Professor David Goldberg is a very illustrious student of Holland, and is the author of “Genetic Algorithms- in search, optimization and machine learning”, Addison Wesley, 1989. It is the, it comes in the Pearson, Indian low priced edition; people who are very serious about optimization, I encourage you to by this book; it is less than 300 rupees, right.

So, this is the, David Goldberg, of course, now it is an independent consultant; Goldberg did his PhD with Holland. So, his PhD topic was, Goldberg is originally a Civil Engineer, but he did his, he figured out the genetic algorithms along with Holland, and he similarly responsible for the spectacular growth of this field - Genetic Algorithm; his student Kedan is a professor in IIT Kanpur; IIT Kanpur is a Genetic Engineering Laboratory, ok.

So, India is the most well-known optimizer or whatever, he is a most well-known optimization expert; of course, he does optimization machine design, he is from design. So, he is well known; he is highest number of citation, he is in the ISI top 100; you know ISI Thomson top 100, most sited scientist of the world in all branches of the engineering. Kalyan is there, Kalyan might there, right. So, he is also written a book, excellent book, small book, I keep telling about this book, “Optimization Engineering Design”; 200, 250 rupees, he wrote it, when he was in his thirties, but still, when he was in still assistant professor, he wrote that book, right. So, it is a very good book.

He has not revised it, but he has come out with one more book called “Multi Objective Optimization”, which is very good. Multi objective optimization is a way forward, because single optimization criteria, single optimization, single objective function are all past, they are all gone. Now, you do not, I mean, you cannot just develop maximum efficiency; you know minimum efficiency, and they are orthogonal; that means, when one goes up, one goes down; therefore, you have to come up with some criteria which takes care of both; then, there should be a penalty for deviating from the individual optima.

So, there are various ideas, various thoughts like Paratoo optimality plot. How do you, how do you factor in both the objective functions? If there is a more than 2, 3, how do we handle all this? So, he has written an excellent book on multi objective optimization, right. You are encouraged to look at it, that is also available; both are available, all the books should be available, which I mentioned should be available, right.

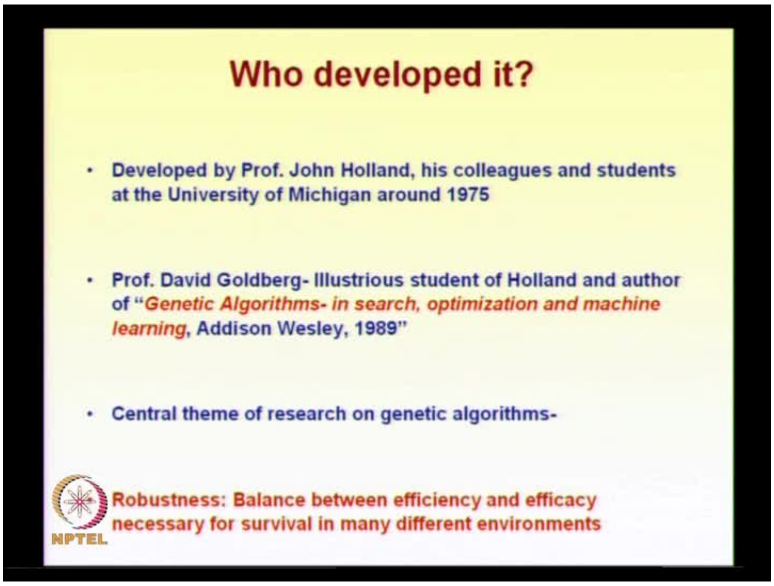
So, what has been the central theme of research and genetic algorithms? Basically, genetic algorithms, there, the central theme has been robustness; if you say something is robust, this machine is robust, the ambassador car is robust, all that, but it gets into a part hole or ditch or something, you can easily come out. But if, if your Maruthi 800 gets stuck, that is very difficult; of course, two people can lift it, that is the different matter; this ambassador, is robust, right. So, it is, it is got a high ground clearance and all that. So, it is designed for mini truck. So, pleasure is not the objective function there; so, that is the robustness.

Here, we are looking at in optimization, robustness is a balance between efficiency and efficacy, necessary for survival in many different environments. Robustness of a species, right, robustness for the human being has arised; how much can we, how much of temperature can be withstand? How long can we go on without having food and water? How much can we survive under hostile condition? How long can we survive under hostile condition? That is basically robustness.

If equivalently, if you look at it from the optimization point of view, how many different classes or problems will it? How many different classes or problems are amenable to genetic algorithm? Or, how many different classes of problem can be solved using genetic algorithm? So, but we are not claiming that, for all these problem, genetic algorithms will solve it better than other sophisticated or specialized optimization techniques.

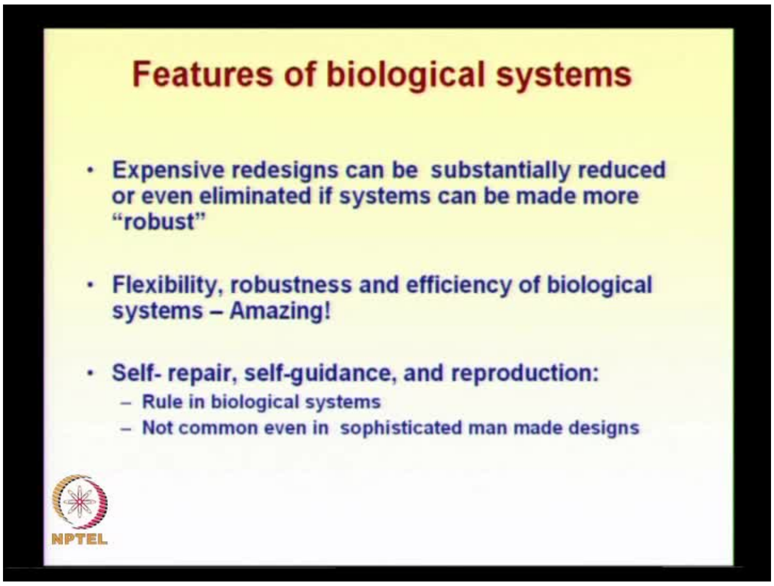
But the FUNDA is, for a white class of problems, genetic algorithms works reasonably well, that is the robustness; for a specific problem, for a specific two variable problem, where it is continuous differentiable; you can get dy by dx, d square y dx square, your conjugate gradient will work very fast; it will defeat, genetic algorithm hands down.

(Refer Slide Time: 05:14)



At the same time, if we apply the conjugate gradient for a function which is got, local, minima, optima, and it is oscillating madly, then the conjugate gradient will just get choked. So, there are certain kinds of problem, where specialized techniques will be superior to GA; there are other kinds of problem, where getting derivatives and all that is messy, or competition it is expensive, and there are other issues. So, for a white class of problem, GA can be applied; so, that is the general theme of research on GA.

(Refer Slide Time: 10:14)



What are the features of biological systems? Expensive redesigns can be substantially reduced, if systems can be robust; that is clear, right. Robust means, it will not break down; even for a slight test deviation from design conditions, if it does not work, it is not robust, right. For example, if we have a very sophisticated air conditioner and if it slightly temperature increase about 40 or 41 degree, suddenly switches off, many times; it is make it 40 in India. So, that is why, they say it as a tropicalized compressor, you know.

For example, air conditioner which works in, for example Europe, will not work in Saudi Arabia. What is the maximum temperature outside in Saudi Arabia? Somewhere it can reach 50, it can reach 50. So, the compressor has to be redesigned, but the same compressor used in Europe, it is not required; are you getting the point? Therefore, so there we say it is robust; if there is a air condition equipment which is robust means, it can work in many different types of conditions, right. So, costly redesigned can be eliminated, if systems can be made robust; they should not be finicky, they should not easily fail, when the design conditions are slightly altered.

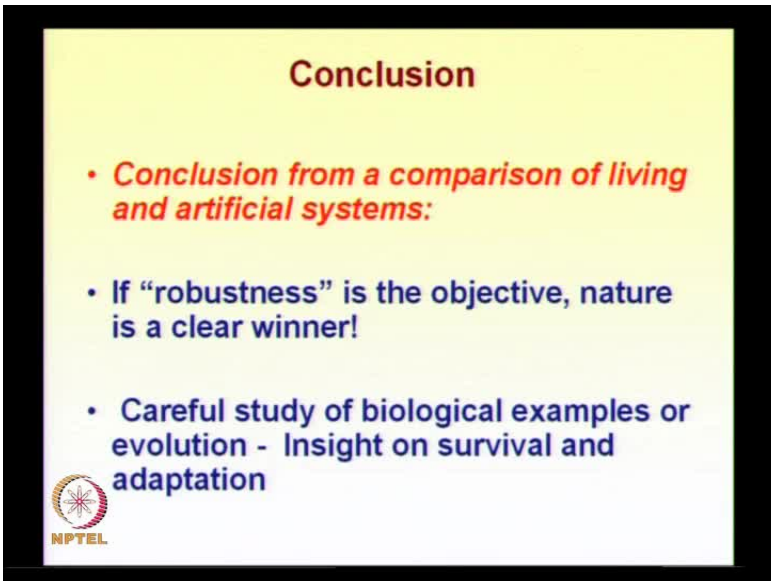
But, if you see biological systems, the flexibility, robustness and efficiency of this system is amazing, right. They are able to adapt, I talked about the mosquitoes; there are so many things, which adopt; their flexibility, robustness and efficiency are amazing. And, self-repair, self-guidance and reproduction. All the dog, tigers, elephants, they do not have doctors, they do not have clinics, they do not have Apollo hospitals and all that; they also get injured, right. They also, it is all self-repaired; there is no anti biotic and no vaccination, swine flu vaccine, nothing is there.

So, therefore, if some species get something, that is it; some will die, only the best will survive; best those which survival where already immune to that. So, when there reproduce and other follows already taken care of the swine flu or whatever, that is the way it works; if you leave it like that in human being also, that is the way it will work, but now we are intervening medically, are you getting this point?

So, and also for example, if you are, if you, if you have fracture, right, you cannot cut this and put it give it to the service center and get it fixed, then. So, while you are doing other things, you are talking, eating, sleeping, you put some sling, you take some Anti Biotic and then this will get heeled automatically, while the other processor are going on.

So, self-repair is out standing in biological system; self-repair is almost impossible in, almost impossible in artificial system. Nowadays, what we are saying is, atleast you gives something mal function, in all that, we can take the look at it. But, self-repair is; so, it is a long way before artificial systems can mimic all of what can be achieved by biological systems. So, this self-repair, self-guidance and the reproduction, of course, thank god, artificial system cannot reproduce. So, we, reproduction, self-repair and self-guidance are a rule in biological system, but they are not common even in sophisticated man made designs.

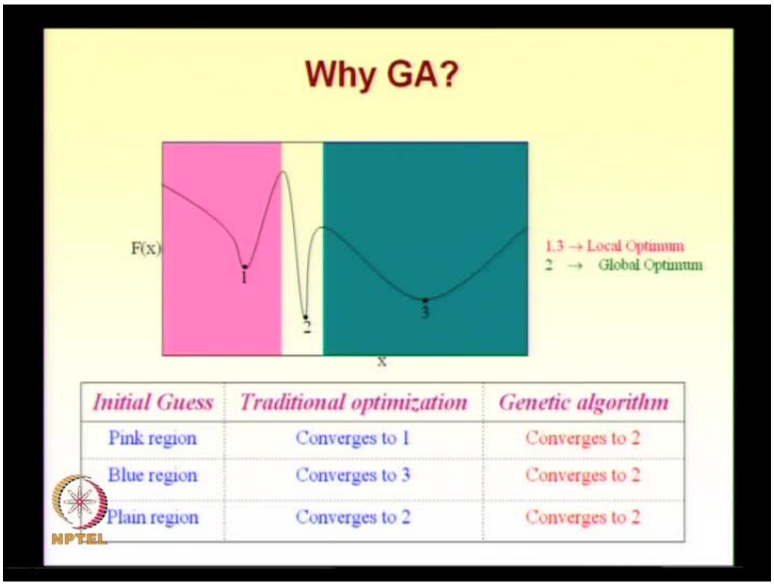
(Refer Slide Time: 13:16)



So, this is not the last slide. So, the conclusion from whatever discussion we had thus far is, from a comparison of living and artificial systems, we can conclude that, if robustness is the objective; that is, you want an algorithm which will work reasonably well for variety of problems, I mean, if you look at evolution, you can draw ideas from that. The basic point is, if robustness is the objective compared to manmade design, natural designs or a clear winner, so careful study of biological examples or evolution gives us, they are very insightful.

If you look at biological example very carefully, you can understand the ideas of how species survive, how they adopt themselves to changing circumstances, and all that. This how do you incorporate it? Incorporate in engineering problems.

(Refer Slide Time: 14:08)



Why genetic algorithms, right. So, is there a pointer? This is the pointer. No, it is going to come on NPTEL, I will proceeded. So, why genetic algorithm? For example, if you look at one variable problem f of x is x, I am taking a function which has got two local minima and one global minimum; the two local minima are 1 and 3, and the global minimum is 2, right.

So, if you have a robust optimization algorithm, robust optimization algorithm is one, which regardless of its starting point will always converge, will always converge to, what is a point here?

Student: 2. Converge to 2, but let us look like this, let us look ahead this way; for example, if the initial guess is in the pink region, a traditional optimization algorithm like a steepest ascent, steepest descent, conjugate gradient, mark or whatever, it will converge to 1; it will declare that one is the solution, because if you go right of 1, it will push it back here; if you go left of 1, it will push it back here. Therefore, it will say that 1 is the true solution.

If you start, if you start with the blue region, a traditional optimization algorithm will converge to 3; but, if you use a genetic, if you use an evolutionary optimization technique or inequivalent global optimization technique, regardless of whether you start in the pink region or blue region, you will get 2 as the final answer. But, now, coming to the plain region, if you start in the plain region, your regular algorithm will also work.

That is, by chance, if you are able to start in this region, that is you are so smart, that you know that the solution here; then maybe you do not require an optimization algorithm. You can just think and say x equal to 5.4. Next, so, only if you, what will happen is, when we are all sitting for your B Tech and M Tech, first the students will, ok. So, the target is always the professor. Now, may be he did not get a good grade or whatever.

So, if you start, if you start in, if you start at the region, in the region 2, both the regular traditional optimization technic and your genetic algorithms will give the same answer. So, the FUNDA here is, regardless of initial point, a global optimization technique should give the correct solution, should converges the correct solution, genetic algorithm will do. We will, genetic algorithm is one such algorithm which can be used, if we have got an f of x like this - point number one.

And, several engineering problems may have local or local minima like this, and thus only one global optimum, and you want to determine the global optimum. Now, this gives, this should give you additional ideas; can anybody say? This picture can give additional ideas.

Student: Global maximum.

Global maximum is ok; whatever is applicable to global minimum, we can apply to global maximum.

But do you think, but do you think the genetic algorithm will be fast or slow compared to conjugate gradient?

Student: Slow. Slow. So, there is a disadvantage, that this will be slow; there is an advantage, that will converge to the global optimum. Can you combine the speed with the robustness? Can you have best of both the word?

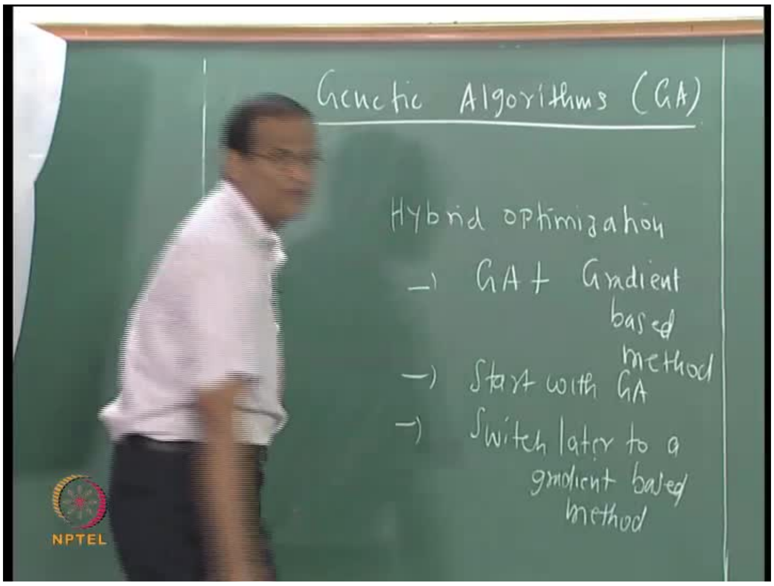
Student: Yeah.

How?

Student: Initially start with genetic algorithm.

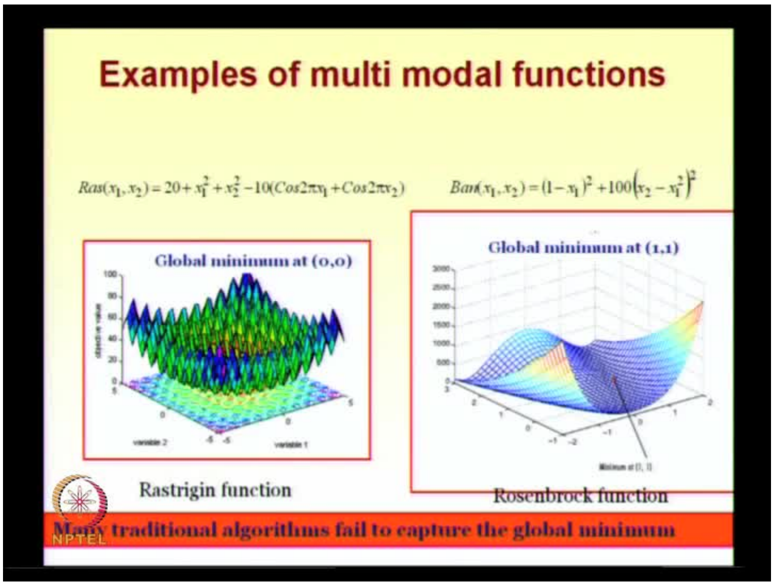
Initially start with genetic algorithm, very good.

(Refer Slide Time: 18:26)



So, you start with genetic algorithm and 0 in, and a white region. Once you come to the white region, then you quickly go to your conjugate gradient or your (()) or whatever. So, this is called a hybrid optimization technique. You start with GA, then switch later to gradient based method; this is routinely used nowadays, for complex optimization problem.

(Refer Slide Time: 19:17)

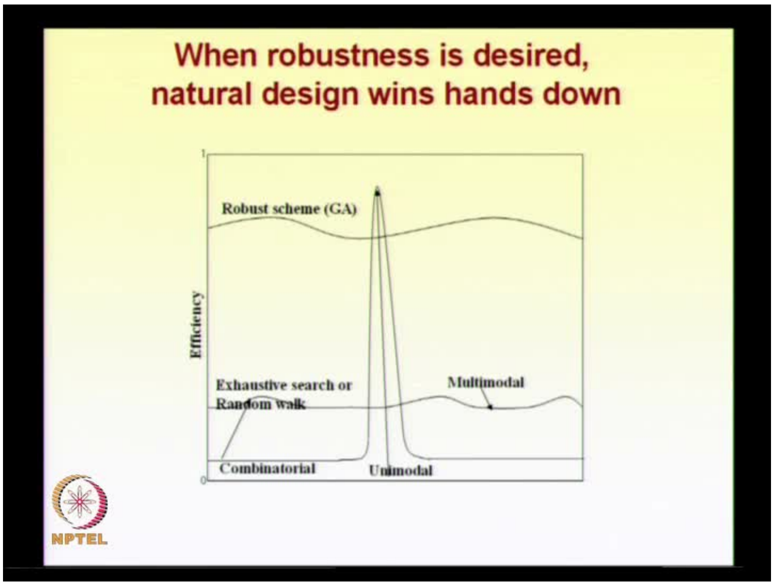


Mathematically, what are there, some examples of multi modal functions? There are several, there are Rastrigin function, which is given by 20 plus x1, x1 square plus x2 minus 10 cos 2 pi x1 plus, cos 2 pi x2, is a function which has got a; it really torchers, why? So, it goes up and down, there are several peaks and valleys, but the minimum of the function is 0, right. I think, its global minimum occurs 00. What is the value at 00? 0, 20 minus 20, 0. Now, the Rosenbrock function is called, also called the banana function, so where 1 minus x1 square plus, 100 x2 minus x1 whole square; so, global minimum occurs at 1 1.

So, if somebody is interested in working on the theory, optimization theory, if you develop a new optimization algorithm, the reviews of journals will expect you to test your algorithm again standard functions like this. And then, you have to bench mark, on a computer, on a particular computer, how many iteration does your technique take? What is the final level of accuracy it reaches? First, it should give the global minimum; and then, what is the speed? What is the time?

So, these are all considered very standard; you cannot suddenly say, out of the box thinking, I developed new algorithm, fine; but, you have to test it again some standard function; just like we say your numerical procedure is to be validated, if you doing ACFD solution or whatever, right. So, these are some multi modal functions.

(Refer Slide Time: 20:55)



So, when robustness is desired, natural design wins hands down. So, if you look at the efficiency of a particular optimization technique versus the kind of problems; I am looking at three kinds of problems- one is a unimodal problem, which has got only one peak or valley; when there is a combinatorial problem, where there is various combinations, for various combinations; there are five machines, each machine can do 5 types of operations; then, there various combination, which machines will do which job, so that you maximize a profit or minimize the cost or whatever. So, that is the combinatorial problem. Multimodal problems are some mathematical problems or engineering problems, where there are several peaks and valleys. So, these are some classes of optimization problems.

If you look at unimodal problem, you conjugate gradient or steepest ascent, steepest descent will be very efficient; so, it hits a peak here. So, the efficiency is almost close to 1, right. But, if you look at, but if you try to apply the same to multi modal function, it may not converge or it may converge very slowly; that means, you have to restart, you have to restart with initial, with the different initial point, right. So, the efficiency is very low. The ordinate is efficiency, the abscissa is a kind of, is a type of problem. For an exhaustive search or random work problem also, the efficiency is very low for a combinatorial type of problem, because our combinatorial type of problem, you may try to exhaustively search different kinds of options.

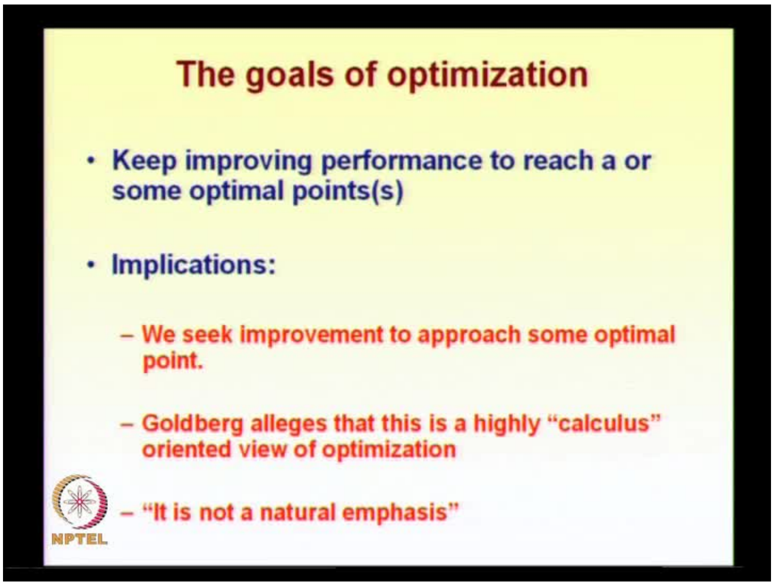
But now, if you look at the G A, GA will have efficiency which is more or less like this. So, the GA has an efficiency, which is far greater compared to what a traditional optimization problem has, for a multimodal problem or what an exhaustive search method have. But, for a specific unimodal problem, it will be lower than a sophisticated algorithm, which is developed for a unimodal function; are you getting the point?

So, that is the robustness; while the exhaustive search or random walk, works within equalitarian low efficiency for all class, for a class of problems, this works with a reasonably high efficiency for a white less problem. But, you cannot claim that, for all the problem, it will be the highest; that is far from the true.

Student: (( ))

Yes, correct, the same machine and this, getting the correct solution.

(Refer Slide Time: 23:18)

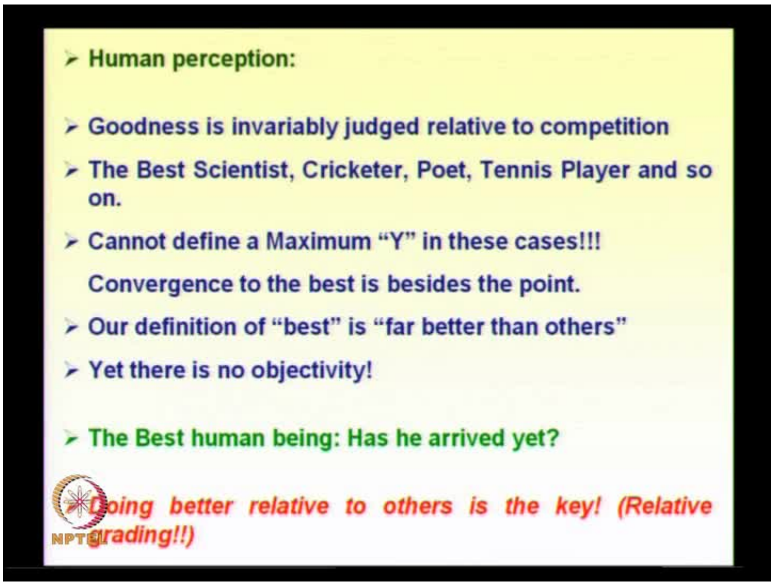


Now, we have to look at the philosophy optimization. What is the goal of optimization? So, we keep improving performance to reach a, or some optimal points, we see whether Y improves, right. If Y keeps on improving, then we are, we believe that, we are going in the right direction, and I will try to get the maximum value.

So, the implications are, if you such an approach, the implications are, we seek improvement to approach some optimal point, right. So, there is always some optimal point; the goal is to reach to that optimal point; that is what we have been taught, right. There is some optimal point, the goal is to reach that optimal point.

What was Goldberg alleges that, this is a highly calculus oriented view of optimization; because, you have learnt so much of calculus, you feel the d y by d y must be equal to 0, somewhere it should be a peak, and both side it should drop sharply; he says, it never happens in nature, it never happens in any engineering problem; it is the. So, he says, it is a highly calculus oriented view of optimization. So, alleges, but that it is not a natural emphasis, why?

(Refer Slide Time: 24:20)



The human perception, goodness is invariably judged relative to competition. So, the best scientist, cricketer, poet, tennis players, and so on, we cannot define a maximum Y for each of these. Somebody's outstanding means, compared to others is very good, is far better compared to other; but, there may be somebody else will come after 20 years is better than this. You cannot put criterion; then, if it is satisfy all these, use the best. Among all the musicians, he is better, far better than other; among all the players, that is all we can.

So, human goodness is, so we cannot define a maximum Y in these cases, convergence to the best is beside the point. Where is the convergence to the best? Who know the best? What is the best? Why are we so hung upon this d y by d x equal to 0? That is Goldberg's, that is the way they started GA, right.

So, our definition of best is far better than others, yet there is no objectivity; there is no objectivity in this, you say somebody is better than this. So, can we say that best human being as we arrive that? How to define the best human being? What are the parameter? Somebody can earn more money or somebody you can solve the equations, somebody you can propose theory, new theory, I mean, any theory can be challenge and can be replaced by a new theory, right; that is, that is how we progress, right.

So, doing better relative others is, doing better relative to others is the key, that is the way we introduce the electric key. So, that is beside the point, that is the tangent, right.

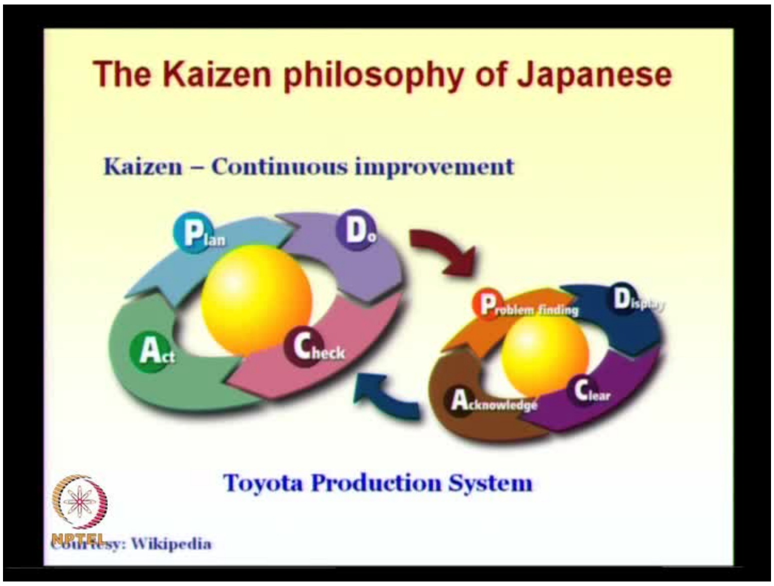
So, but doing better related to others, is very important. Therefore, let look at whether the optimization algorithm goes in the right direction; after it has reached some particular level, stop it; do not try to get one optimum solution, that is what we say.

(Refer Slide Time: 26:15)

So, from the perspective of a genetic algorithmist or a GIA specialist, the priorities of optimization have to be relooked. The most important goal for optimization is improvement; improvement is more important, how the objective function changes? So, the key question is, instead of optimizing, use something called "satisficing"; I put it in double codes, because it does not exist in the English; it is not there in the dictionary.

So, can we get to some good satisficing level of performance quickly? Can you get to some? So, four design, we are use special temperature all this; with the combination of this variables, can we reach a good value of efficiency for the power point? Right; so, that is the goal. So, attainment of optimum is less important for complex systems.

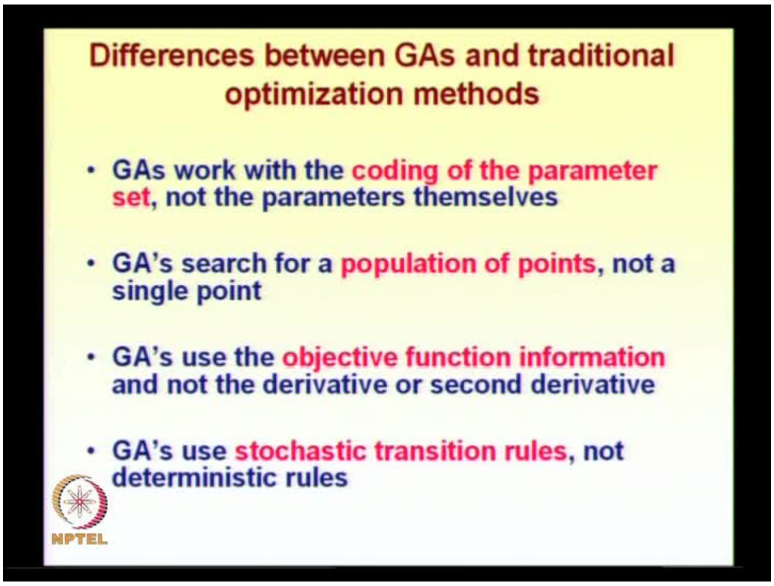
(Refer Slide Time: 26:59)



In fact, the Kaizen philosophy of Japanese; are you heard about this Kaizen? Kaizen is continuous improvement. The middle level workers, Japan strategy is, you should not have separate section of supervisor; somebody will make, somebody else will check, they got rid of this. The fellow who makes is also the fellow who checks; he also reports the deficiency is this and then, and then, there is a self-correction; so, there is a continuous improvement. So, you cannot see rapid progress.

But, because there is an incremental progress towards, over a period of time, you can see substantial progress. So, Toyota uses this; so, this basically from Wikipedia, Toyota production system is act planning; if I spend too much time on this, if this, like a dooms class, ok. So, but the quick point I want to make is, the Kaizen philosophy of the Japanese is also something like this; continuous improvement in small steps.

(Refer Slide Time: 27:54)



What are the differences between GAs and traditional optimization methods? Do not be too happy that I cannot ask questions in the exam, because it is all theoretical, because we are going to solve a problem now. So, GAs work with the coding of the parameter set, not the parameters themselves; generally, the parameter, the variables, x1, x2, x3, are usually replaced by their binary equivalent. So, you replace them in, into 0s and you convert them into 0s and 1s. It is possible to write a genetic algorithm code, without this binary representation also, but the most popular is that with the binary representation.

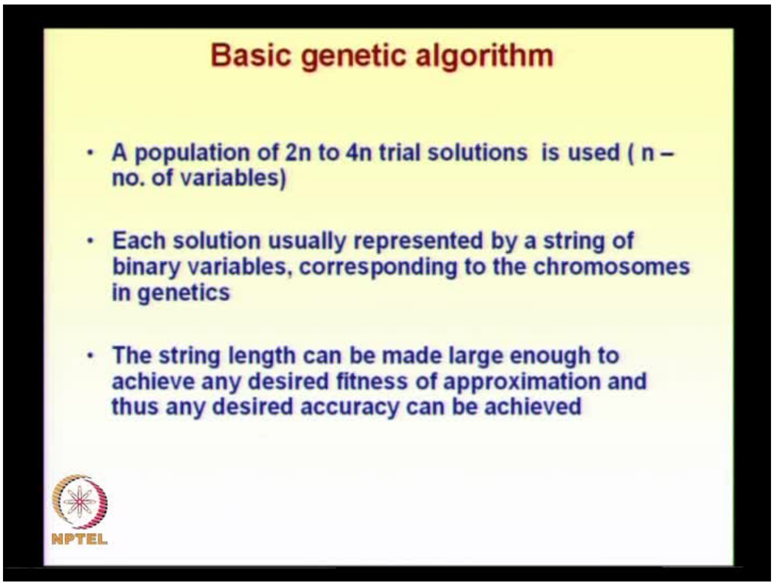
So, the GA search for a population of points, not a single point. Instead of taking a single value of x1, x2 and finding Y, and seeing how Y changes with x1 and x2; if it is a two variable problem, you take x1, x2, x1, x2 A; that is, A is 1 particular value of x1, x2. So, you take A, B, C, D, they represent different values of x1, x2. For A, B, C, D, you calculate Y of A, Y of B, Y of C, Y of D; you find which is maximum, which is minimum.

Convert x1, x2 into 0s and 1s, into bits, and then, you mix and match the bits from the better parents; the better parents are, the better parents are those Ys which are higher. Then, produce new values of A, B, C, D; for new values of A, B, C, D, A, B, C, D means what? A is x1, x2, B is another x1, x2. So, you produce children, which then become parents; then, there is a strategy involved, that among all this parents, only the fittest can survive and reproduce; this what happens in nature. So, you mimic nature there. This is essentially the genetic algorithm, it become clear when we solve a problem.

So, GAs, so it searches from a, it search as a population of point. So, it take 4 or 5 points, and keep track of how this, the number of members in a population, in a sample is kept fixed, that is 1, that is the usually popular method of GA. So, if you take four or five solutions, how these 4 or 5 solutions evolve with generations or iterations? After certain stage, you can either take the average of all these four and say that is the average optimum; that is the optimum which is eventually reached. You will reach a stay, where there is little different between the members of the species; that is the way it should be, right. And there is the continues improvement, all should be equally strong or equally weak; that is what GA stripes to HE.

So, GA is used to objective function information; we do not use the information of the derivative or second derivative; we use only the, we use only the information on the objective functions. So, GAs use stochastic; stochastic means probabilistic transition rules, not deterministic rules, right; that is why, it is a stochastic optimization technique. So, we use some random numbers, and stochastically, we determine certain rules; this will became clear in a little while.

(Refer Slide Time: 30:51)



So, what is the basic genetic algorithm? Anyway, moodle we have created something for this course. I converting into PDF, and you do not have to take it down. So, and for the purpose of exam and other things, anyway in Monday's class will work out some other; we will work out a problem from start to finish, we will do 1 or 2 iteration. So, that will be there in notes; it should be helpful for your exam, right. So, just listen, so basic genetic algorithm.

Yes.

Student: (( ))

Yeah, Monday first hour, we will do, we will work out the problem GA; if you want to submit, that is all to welcome, but thinks you are, you may not have prepared for that. So, 1 to 1. 50, we will work out the problem in GA, then 2 to 2, 50 you will do the assignment test, assignment test will cover search methods, after Lagrange multiplier.

Student: (( ))

Yeah, they are already every class; we cannot say this form this last class.

Student: (( )).

When? Which exam?

Student: (( ))

But, it is 3' O clock know. So, you can finish it by, you can finish it by 2. 45.

Student: (( ))

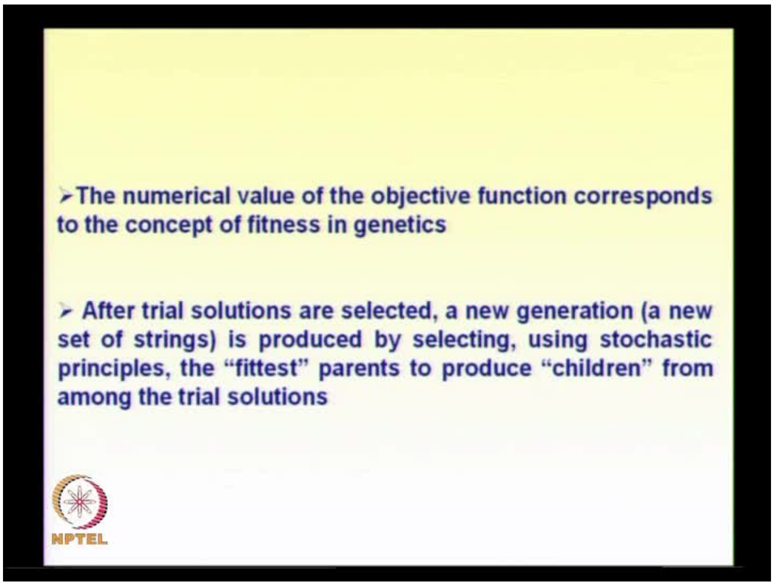
Which one?

Student: (( ))

No, 12 to 12.50 other people, you are always thinking that only the B Techs were taking the course. So, please expand your objective function; there are M Tech, MS, PhD and degree, so 5th years students, 4th year students. So, it is very difficult to change us; it is a very heterogeneous crowd. If it is, all the students are your classmates, it is no problem; for example, he is from E D, your friend next to you. So, I have 6 M Tech students here, and the lunch breaks varies, differs from B Tech to M Tech and there are lot of issues; it is very difficult to change anything.

A population of 2 n to 4 n trial solution is used, where n is number of variables. So, if it is two variable problem, you basically use 4 to 8 solution; you start with 4 to 8 solution, each solution is represented by a string of binary variables, corresponding to chromosomes in genetics. So, each solution, you represent it has a 0s and 1s; the string length can be made large enough to achieve any desired fitness of approximation, that is even decimal can be taken care of, I will explain it you shortly. And thus, any desired accuracy can be achieved.

(Refer Slide Time: 33:12)



Then, what is the objective function? The numerical value of Y, corresponds to that, corresponds to the fitness using genetic; the fitness person will, the fitness member of species will survive and procreate, that is the Darwinian theory.

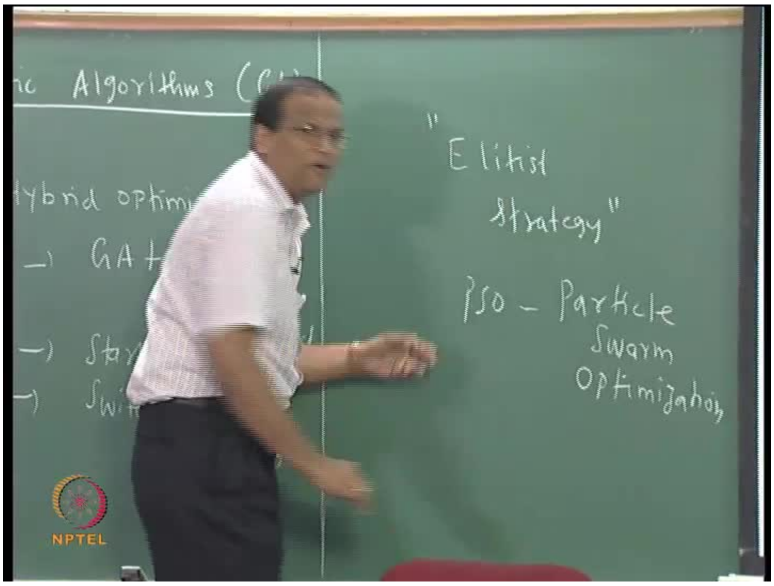
What is that fitness in genetics? The fitness term in genetics is, equivalent to objective function value here, right. After trail solutions are selected, a new generation is produced by selecting, using stochastic principles; the best parents, the fittest parents, to produce children from among the trial solutions.

So, in each child, you mix and match the bits from two parents, and then produce new children. And then, again convert this 0s and 1s back into Y, into x1 and x2, x, x1, x2, x3, x4; it is a 4 variable problem. With new values of x1, x2, x3, x4, you can calculate the value of Y. Then, again put them in ascending or descending order; if it is maximization problem, choose those values of these parameters the design variables, which give the highest values of Y; this will become clear, if you solve a problem.

Sometimes what you do is, some random alteration of binary digits in a string produces the advantageous and sometimes disadvantageous effects of mutations. Sometimes what happens is, somebody is already having a high fitness, somebody is already very healthy, is already having high fitness, but because you always want two parents to mate and produce children and all this, what will happen is, half the chromosome will be cut and half the chromosome from some other parent will be joined; and it may so happen, that is new child may have sometimes an efficiency is poor, because already that fellow the combination of bits is such that, it is high, it is having very good quality.

So, this crossover, repeatedly this mixing and matching, and cutting and pasting, will result in a drop of quality; you randomly do what is called mutation, that is, in 1 in 20 bits, you change 0 to 1 and 1 to 0. So, this is a random and it is stochastically done; that is, it is randomly done, so that sometimes undesirable effect of this crossover, crossover means cutting some chromosome, and then, in the two parents, and then mixing and matching to get the new children, that may, that may lead to some trouble.

(Refer Slide Time: 35:30)



So, genetic algorithm is, have also developed some other techniques to overcome this. What is, the other technique is, what is called the elitist strategy. What is elitist strategy? I hope by now you got an idea of the algorithm; if there are 4 or 5 solutions, that is, if it is only x1 and x2, so you take x1 and x2, 5 different combinations. And, if it is a maximization problem, rank Y1, Y2, Y3, Y4, Y5; you discard Y4 and Y5. Between Y1, Y2 and Y3, that corresponds to some value of x1, x2; you represent x1, x2 as binary.

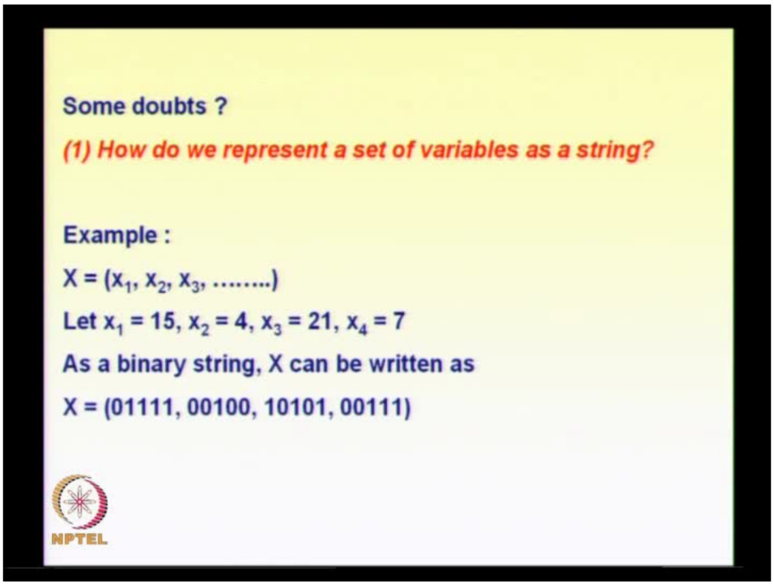
Then, you choose, you choose a corresponding mate, 1 will have a mate 2, and 2 will have a mate 3 and so on, like that you choose, and then mix and match and produce a, produce a new generation, right. What happens is, sometimes the best ones will, because of this crossover of chromosome, the best one sometime may lose its quality.

In the elitist strategy, what they do is, out of these n solutions, the best solution is left untouched, and the best solution automatically goes to the next generation. And, all these permutations, combinations, mutation, crossover, all these are done for n minus, n minus 1 solutions, where this king remain; the king is the, it is elitist strategy, e remains, but we cannot remain king forever.

Now, if you do mixing and matching, and this n minus, n minus 1 parents will recombine and produce new children, so this n minus 1 parents from n minus 1 parents, you get n minus 1 children; this fellow will also go. Now, he will be ranked; if he is not king, he will be subjected to mutation and crossover again. The king in a particular generation will not be touched, that is a elitist strategy. This is used in what is called particles swarm optimization, right.

There is one more technique, which is called particle swarm optimization technique, is also a technique which is very similar to the genetic algorithm; it is also an evolutionary technique. Where, for example, birds, when they are moving, if there is a flock of birds which is searching for food, then the birds will move, they will adjust their orientation such that, such that each of the birds is at the smallest possible distance with respect to the leader. And, the leader is one who is closes to the food. And, this may keep, and after 15 minutes, the position of the leader may change; are you getting my point? So, they will try to optimize their distance from the leader, and the leader is one which is closes to the food, and it is equivalent to the elitist strategy.

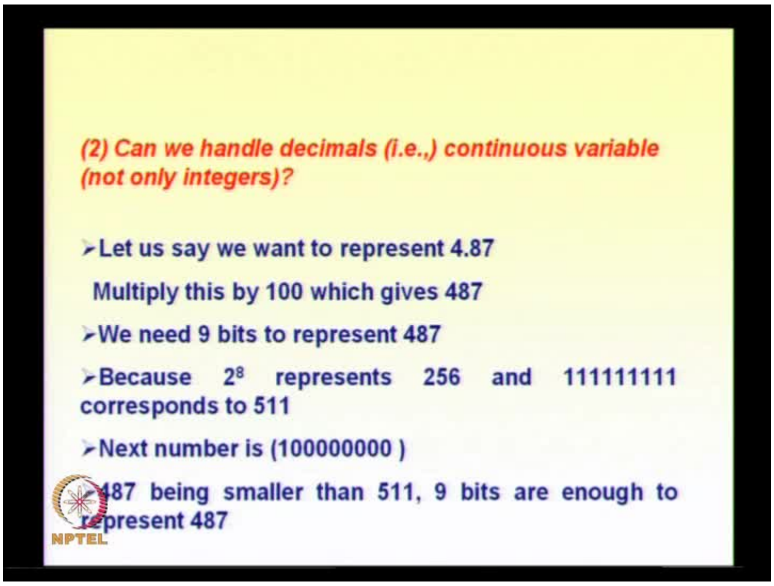
(Refer Slide Time: 38:14)



So, you have some lingering doubts; how do you handle decimals and this thing? And how do we handle a multi variable problem? And so on. So, for example, if you have got x1, x2, x3, x4, 4 variables, and these are 15, 4, 21. As a binary string, you know that 15 can be represented as 0111, 1248, ok; 8 plus 4, 12, plus 2 14, plus 1, 15, correct.

So, what you do is, you represent this as 0111; you represent it as a continuous string; remove the commas. In your computer program, you know that the last 5 bits represent x 4, the next 5 bits represent x 3 and so on. Then, you can. So, each solution, each X, each X is a designed vector; what is a designed vector? It is a combination of the design variables; here, it will be represented as 0 and 1. So, you play with 0 and 1, rather than 15, 14, 21. How do you handle decimals?

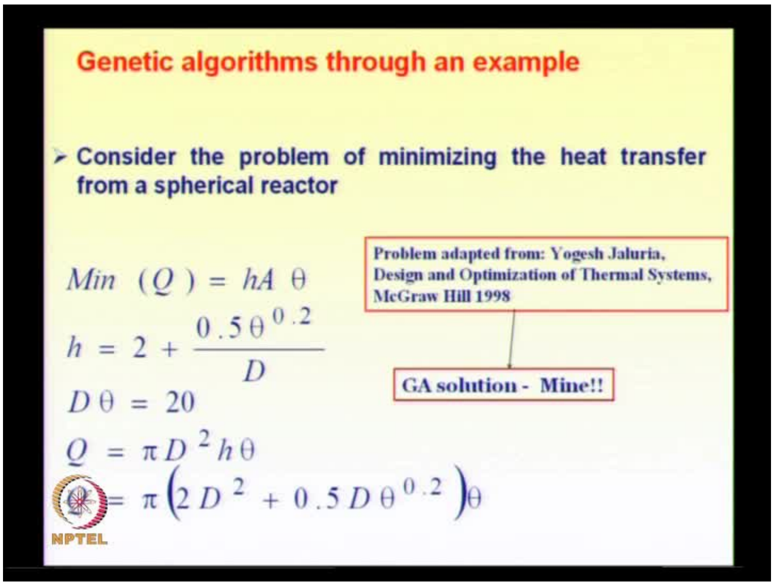
(Refer Slide Time: 39:13)



Can we handle decimals? Yes. Suppose we want to represent 4.87, we multiply it by 100, it gives 48. We need 9 bits to represent 487, because 487, 2 to the power of 8 isto 256, and whatever I indicated corresponds to 511. So, 487 being smaller than 511, 9 bits are enough to represent 487; you represent 487 by 9 bits and say that this represents 4.87. But, with understanding that, when you convert this 9 bits into decimal, you have to divide it by 100, that is all; are you getting the point?

So, if your, the number of the decimals, the accuracy you seek, keep increasing, the number of bits required will also increase dramatically. It just does not depend on the bigness or smallness of x1 x through, how many decimals you want, that also matter. So, if you are sure that a variable lies between 1 and 5, and we need 2 digit accuracy, 9 bits are more than enough; that is, if you want to represent 4.87, 9 bits are enough. As our accuracy required increases, the number of bits required will increase dramatically.

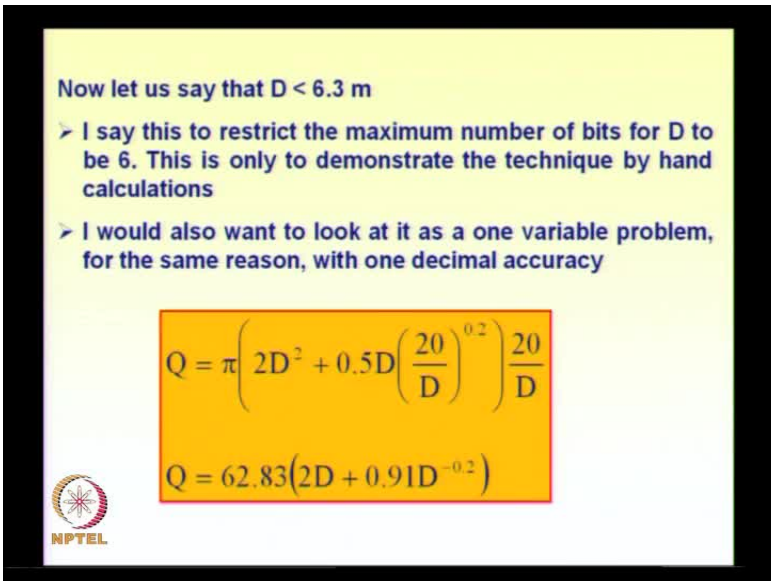
(Refer Slide Time: 40:28)



Now, let us consider the heat transfer from spherical reactor, the problem is taken from Yogesh Jaluria. We will work out a new problem on Monday; of course, GA solution is mine, but I would not acknowledge, this problem is from Jaluria, right, ok. So, let us say that we want to minimize a heat transfer, is the convective situation, minimize h A, h A theta; for this sphere, area will be 4 pi r square, theta is a temperature exist. Heat transfer co-efficient is written in terms of diameter and theta, this is given. And, there is also a strength condition, which says that D theta equal to 20. This problem can be eminently solved using the Lagrange multiplier, or you can convert it to single variable problem and solve it using calculus, but we want to demonstrate how the genetic algorithm works.

So, I can substitute for the theta in terms of 20 by D, and say, Q is equal to h A into theta, pi D square is theta. I can do all these mathematical manipulation and convert it to single variable problem. I can convert it into single variable problem in D, right; this can be easily done.

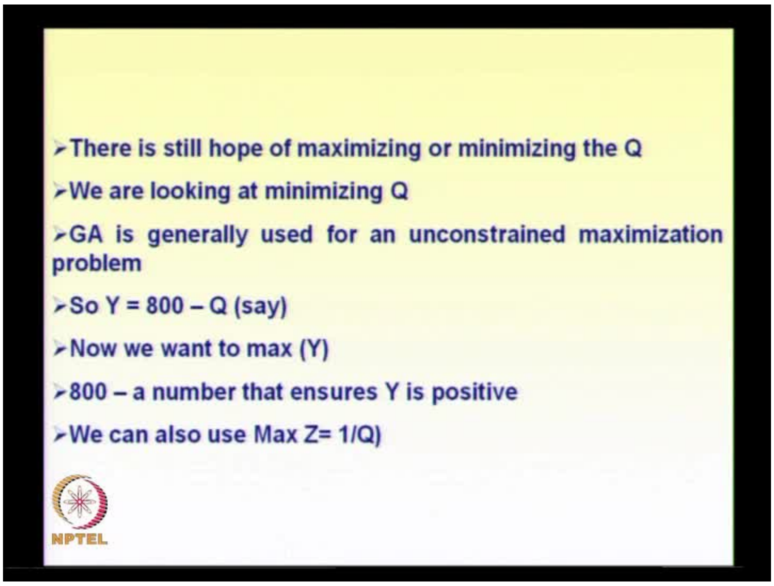
(Refer Slide Time: 41:41)



Now, if you see the first term is D to the power of 1, the second term is D to the power of minus point 2. The first term increases with D, the second term decreases with D. So, there is a hope, that there will be a particular value of D, at which Q will be a minimum; it will be an extremum, while it is very unlikely that extremum will be a maximum, extremum will be a minimum, right.

Now, let us say that, I want to restrict a maximum 6.3 meters, because I want only one decimal accuracy. And, 64, so many bits are required for 64. So, I say this to restrict the number of bits to 6; so, this only to demonstrate that technique. So, I will look at it as a one variable problem, and I will see how genetic algorithms can be applied to this; please pay attention to this.

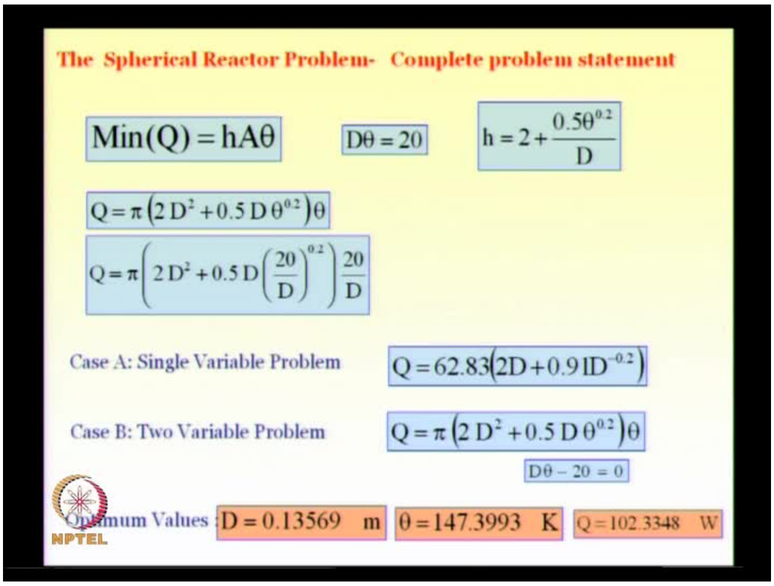
(Refer Slide Time: 42:05)



Therefore, since the first term increases with D and the second term decreases with D, there is a hope of optimizing, minimizing or maximizing. Generally, GA has been developed for maximization, because the fitness continuously improves. We can minimize also, but I want to use the maximization. So, I will say, Y is equal to 800 minus Q. I know that, for various combinations of D, Y will not exceed 800, so that Y become, Q will not exceed 800, so that y become negative. So, I will take 800 minus Q as a Y, is that ok?

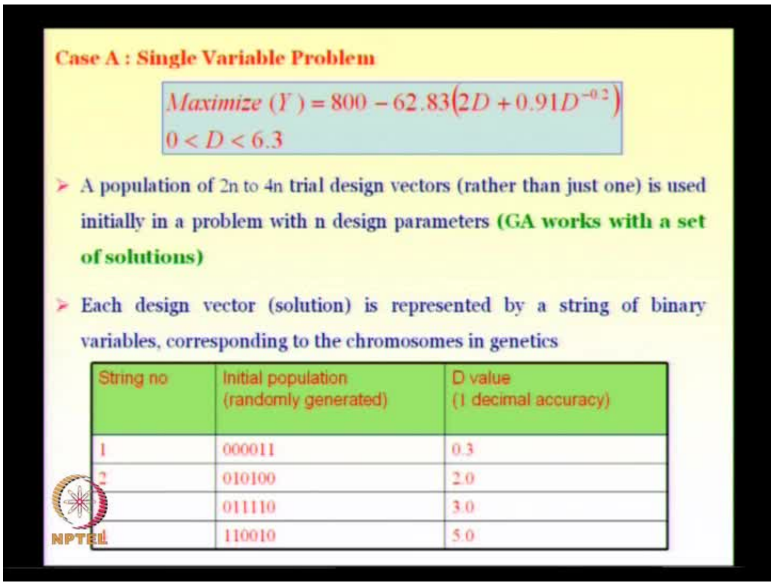
Or, I can take, maximize 1 by Q, but if you maximize 1 by Q, I will have lot of decimals and all that; you can take. So, Y equal to 800 minus Q or 1200 minus Q, whatever you want. Ensure that, it is a positive quantities; is everybody, is everybody with me? Minimizing Q is maximizing some, a minus Q. Now, we want to max Q; we can also use max Z, max Z equal to 1 by Q, that is also possible.

(Refer Slide Time: 43:05)



So, this spherical reactor problem- complete problem statement, heat transfer coefficient variation, strength criterion, Newton’s law of cooling, single variable problem it is like this, two variable problem it is like this; by calculus, the optimum solution is 13.6 centimeter, theta is 147 kelvin and Q is 102 watts. We also know the final solution which we have to get. Now, let us move to GA.

(Refer Slide Time: 43:25)



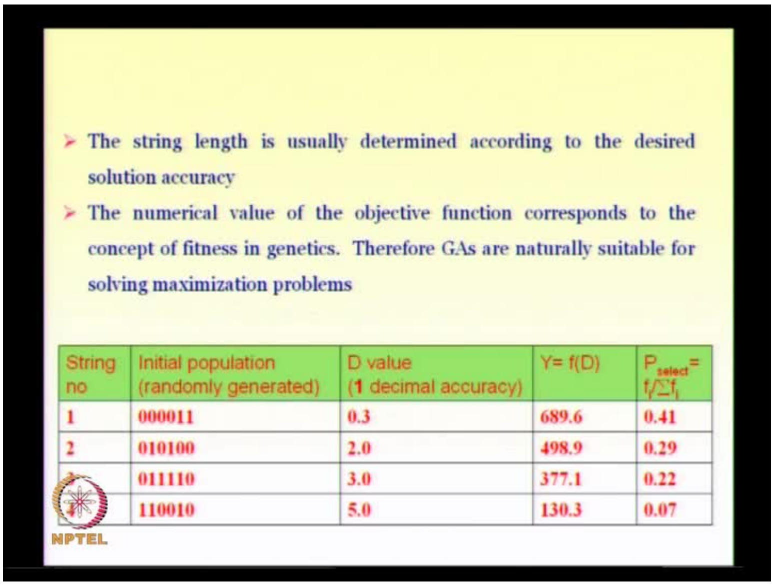
Initially, a population of 2 n to 4 n design vectors are chosen, n equal to 1; n is the number of variables, so I choose 4 n, I took 4 solutions. I am taking 4 solutions. I told you that the diameter varies from 0 to 6.3 meter, right. So, I will take 4 solutions arbitrarily, which are in the interval 0 to 6.3. So, I take 0.3, 2, 3 and 5, so that there is no bias; it is uniformly, I do not take close to 6.3 or close to 0, right. So, 0.3, 2, 3 and 5. So if you take 0.3, it is, the binary representation is 00011.

Actually, if we convert all these binary into decimal, we have to divide by 10 to get this; are you getting the point? Is everybody with me? So, this is, 1 plus 2, 3 meter, divided by 10.3; this is 20, this is 16 plus 420 divided by 10 is 2 meter, right. Now, how do we know that there are no biases in our initial solutions? Visually, I can see that, 0 to 6.3, I equally space. I can also see the number of 0s and 1s in my bits; there are 24 bits or that 12, 12 bits approximately for 0s and 1s, it should have either 13, 11 or 11, 13 or whatever.

So, the other way is, you can generate this in a different way; you can toys a coin, you can toss a coin, you can toss a coin, if it is head, it is one; if it is tail, it is 0; toss a coin, head, tail, head, tail, head, tail, 6 times you do, you will get the first. Then, head, tail, head, tail, head, tail, do not do this in the exam coin. So, you can do head, tail, but this can be a program, right; generate the random number, greater than 0.5, 1, less than 0.50; I will give you random number table, do not worry. You can do that or you can say that I have done like this and convert it back, I have no problem, right. Now, number of 0 is equal to number of 1.

So, it took 4 solutions; among this 4 solutions, you have to rank them, according to the fitness; according to the marks, they have obtained. How you decide the marks? That function is there, 800 minus Q. Q is something into 162.3 into D something, right.

(Refer Slide Time: 45:32)

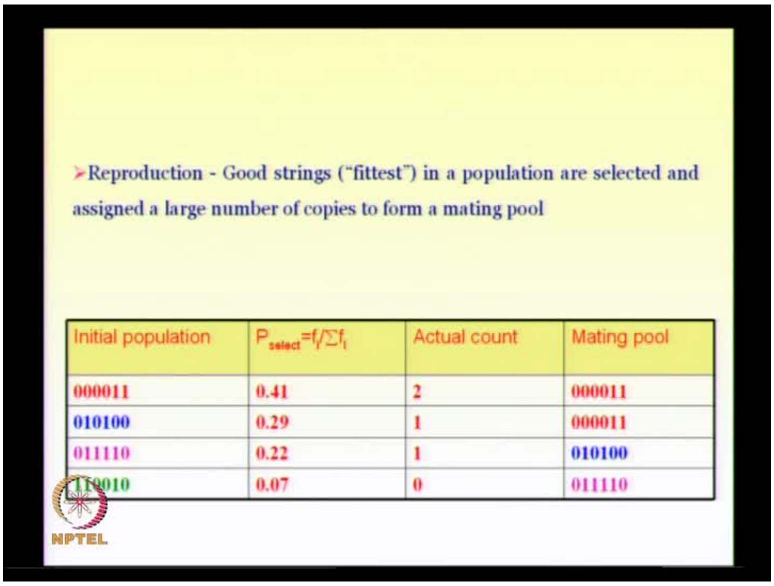


I think it should through this, ok. The string length is usually determined according to desired accuracy. Now, for 0.3, the Y is 800 minus Q into 689.6; somebody with calculator can calculate, no; or, you are accepting whatever I say, fine; so, 0.32, so 689, 498, 377 and 130; watch this very carefully.

I have a last column; so, this column is clear; string number 1, 2, 3, 4, or 4 candidate solutions, the D value and decimal; in binary, this is the value of Y, fine. Now, what I do is, I add all the values of Y, that is, sigma Y, right. And, I divide a particular value of Y, f divided by sigma f, or y divided by sigma f, that gives the relative fitness; are you getting the point?

So, the first one is 41, the second is 29, third is 22, fourth one is 7, the total will be 100 percent or 1.00; that means, the fitness candidate is 1, followed by 2, followed by 3, followed by 4, are you getting the point? Now, only the fitness are allowed to survive and reproduce, that is the genetic algorithm, that is the Darwinian theory.

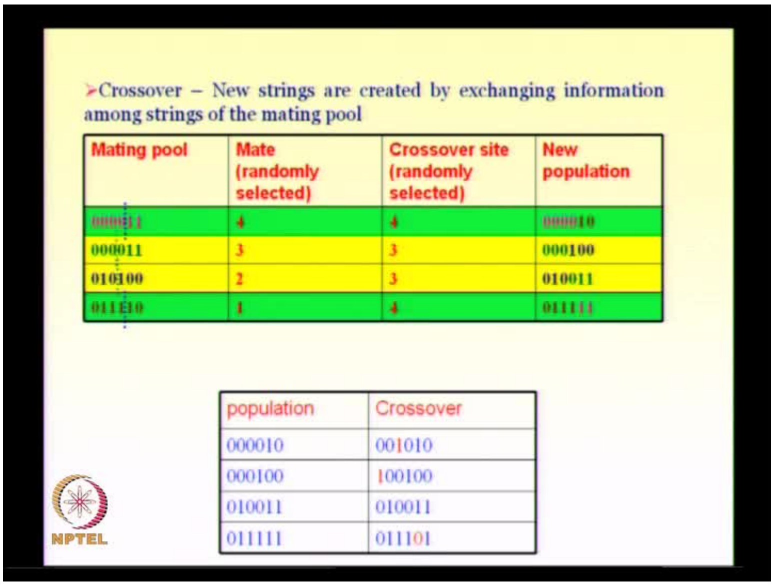
(Refer Slide Time: 47:22)



So, there is an initial population; now, I have to choose a mating pool. So, because this is 7 percent only, relative fitness, it is very bad; I do not have to count it in mating pool, I eliminate him; this 41 is very strong; so, I will represent him, represent him twice, it is like that. So, I representing him twice; this fellow is 1, and this fellow is 1. You should not put all 41, you should not put all the 4 as this, then it will reproduce the same string, you will not get the converge solution. You prematurely converge to this, that is all; are you getting the points; where it is the spice of lies? So, you should be, there should be a, there should be mixing and matching, so let us that you get some change in the chromosome.

Now, if you see this, whatever is red, it is comes twice; look at it carefully, whatever is red is coming twice; blue comes 1s, and then pink comes one, this is my mating pool. From the mating pool, I will get the next generation of 4th children; the green one has been packed, because he is having very poor quality, fine.

(Refer Slide Time: 47:57)

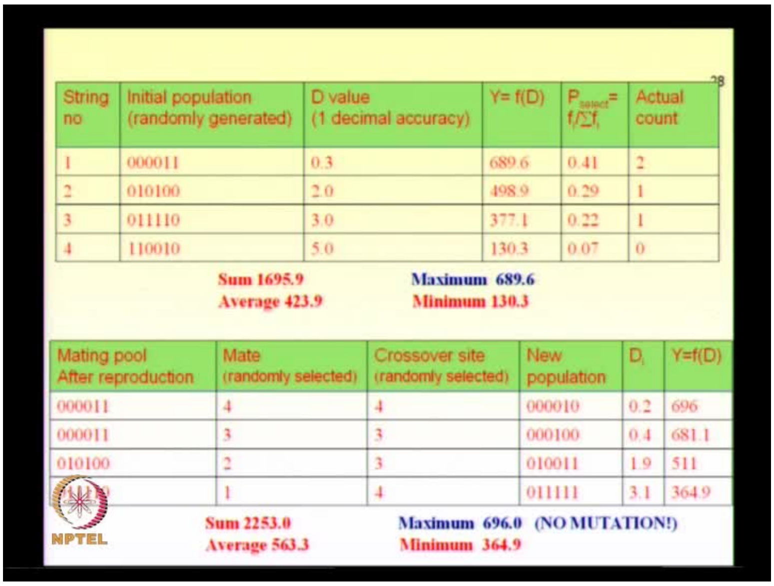


Now, from the mating pool, I have to randomly select a mate. So, for this, the mate can be 2, 3, or these 2 are the same; 1 and 2 are the same. So, we should not take the mate for 1 and 2, because, then it will produce the same thing; you do not want that, that is not allowed; therefore, this has to be 3 or 4. So, I have, we have, lets us say that, we are, we are choosing 4. We are choosing 4 for this; if we choose 4 for this, then automatically for this one, 1 will became the mate, right. For this, you have no other choice, it is only 3, and then it is 2.

Therefore, now you have done two important steps. First, you decide mating pool, based on fitness; and second, you randomly choose, now your choosing randomly the mate; upto this, it was deterministic. Now, it is slowly becoming stochastic. Now, you have to have a crossover site. Crossover site is, after how many digits you want this crossover, that is mixing and matching; that is also arbitrarily decided. Now, let us say that, for 1 and 4, I choose the 4th place; that is, after the 4th place, I introduce the crossover; for 2 and 3, after the third place, I introduce the crossover. What does it mean? Do not get confused. Let us look at these 2, right.

So, when I put a crossover like this, a new population will be 000010; the other one will be 011111, right; this will be the new think. So, pink and dark brown, dark brown and pink, this is the new population generated by these two. For these two, green and blue, 000100, 00010; so, it will be first green and blue, blue and green. So, we now got four new members of the population. Now, it is possible for you now, to convert this back to decimal; calculate the value of Y; and, from the new value of the Y, find random accordingly and proceed. Now, after this one iteration, let see what has happened to the candidates.

(Refer Slide Time: 50:03)



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

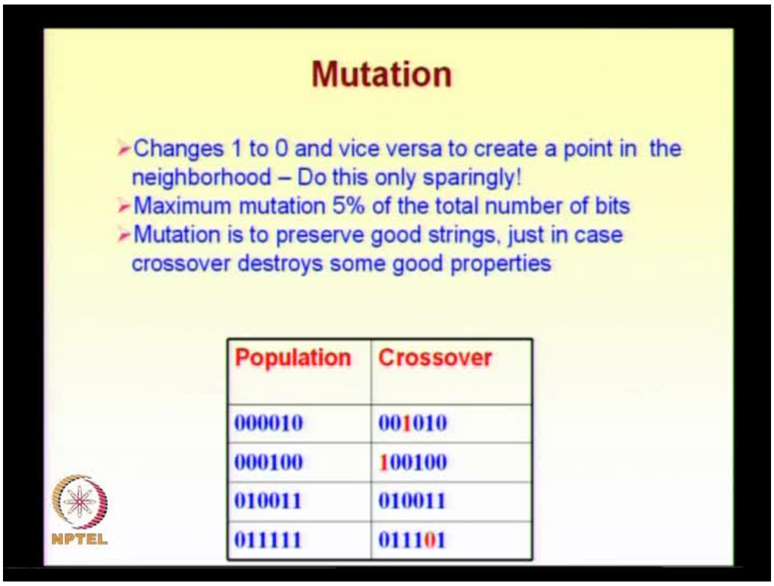
Sum 1695.9
Average 423.9
Maximum 689.6
Minimum 130.3

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
Average 563.3
Maximum 696.0 (NO MUTATION!)
Minimum 364.9

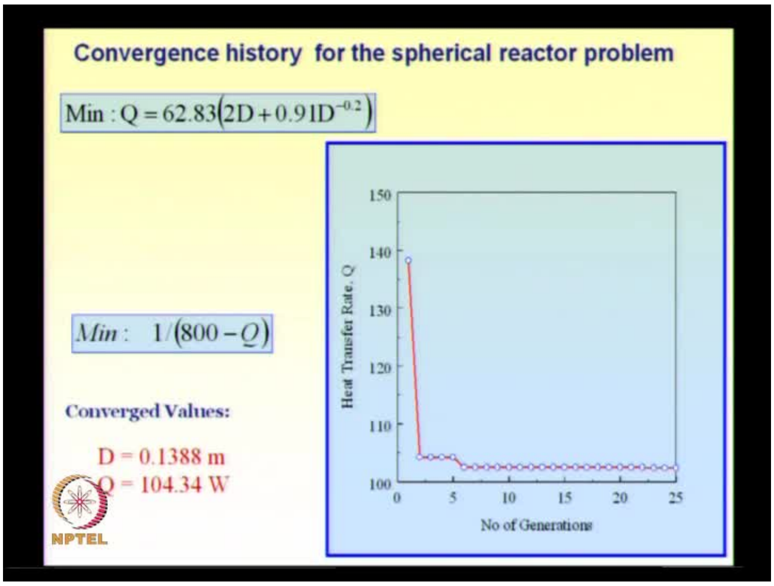
Now, so the initial population was like this. And, this was the table which you already prepared. Now, the new population is this. Now, I can calculate back. This is 2 divided by 10.2 meters. So, you can calculate the 4 new diameters, you can calculate the 4 new values of Y. Now, the first generation, the maximum was 689.6, maximum value of Y, the minimum was 130.3. But, using the Darwinian Theory, using the genetic algorithms, while the maximum did not change much, the minimum was substantially improved; therefore, the average was 423.9, the average short upto 563.3, in just one iterations. So, the average will rapidly improve in genetic algorithms. Finally, all the member of the populations will be having more or less the same fitness, because you have not used the elitist strategy, some good strings can get last, we can introduce mutation.

(Refer Slide Time: 51:08)



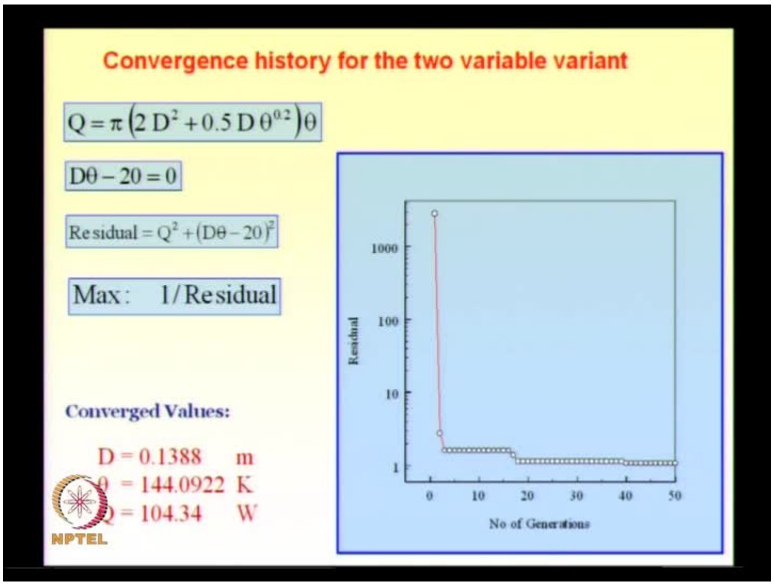
What does a mutation mean? You can change 1 to 0 or 0 to 1, but it has to be sparingly done. So, that means, the maximum mutation is 5 percent of the total number of bits. Mutation is to preserve good strings, just in case crossover destroys some good strings. So, for example, arbitrarily, some where you can put 1 1; I put 3 out of 24? I should not put this; I have, out of 24, I have allowed only 1 or 2.

(Refer Slide Time: 51:39)



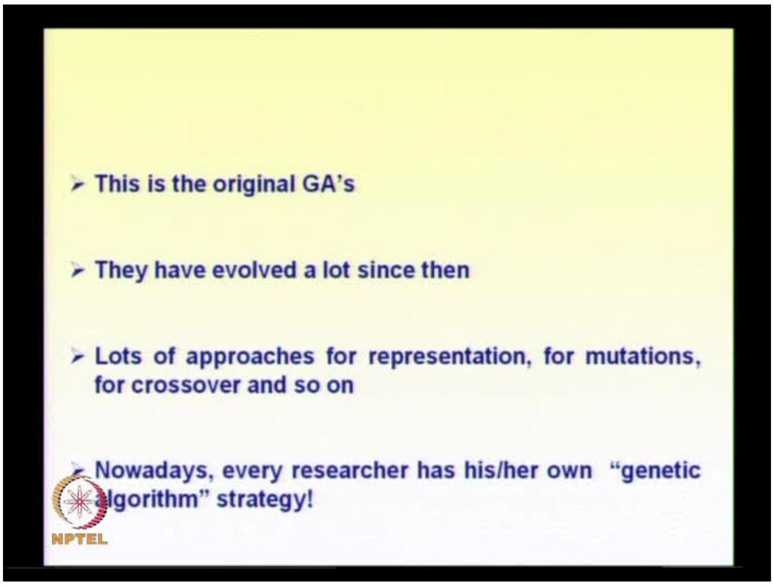
So, now this is the convergence history for the spherical reactor problem. So, this is the heat transfer rate and then how it proceeds; within 4 or 5 generations, we programmed it and we got the solutions likes this; this is for single variable problem.

(Refer Slide Time: 51:49)



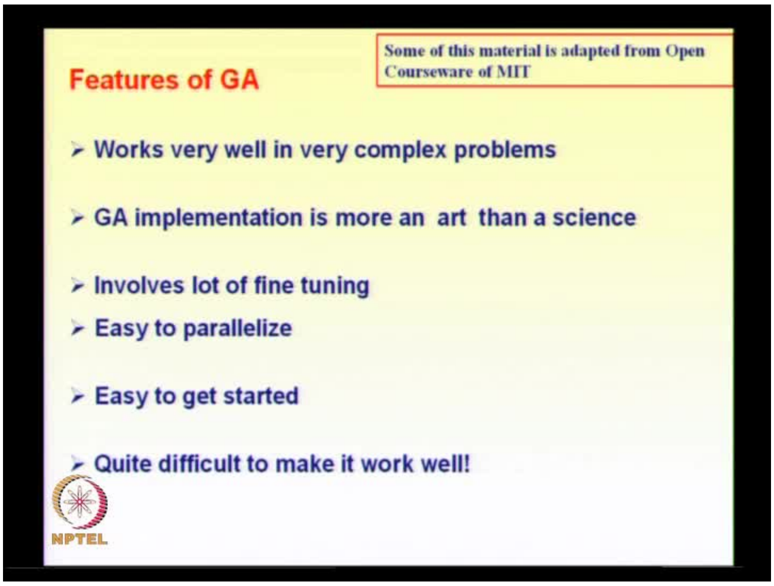
For the two variable problem, it is likes this.

(Refer Slide Time: 51:58)



So, this is the original GA. They have evolved a lot since then. So, lots of approaches for representation, mutations and so on. Nowadays, every researcher has his own genetic algorithm strategy; we do not know what is GA and what is not.

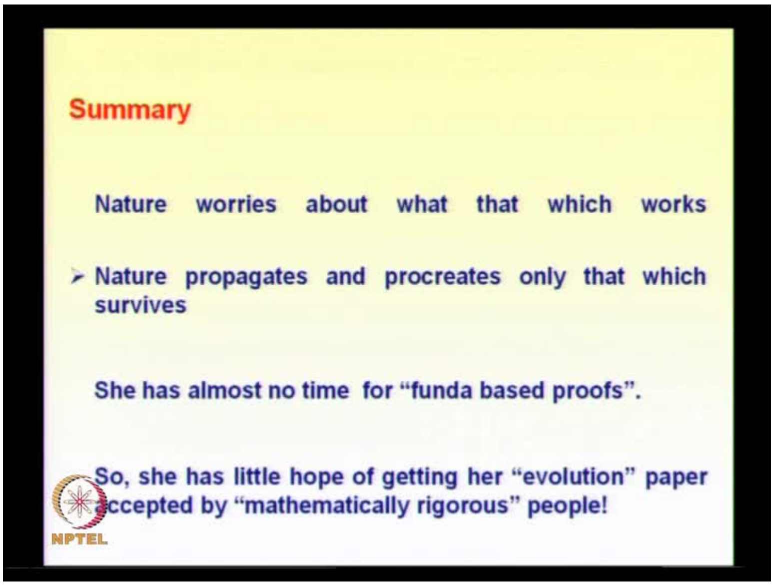
(Refer Slide Time: 52:08)



Some of this material is adapted from OCW. You know, OCW? Of MIT, Open Courseware. So, the features of GA, it works well in very complex problems. The implementation is still an art, because lot of subjectivity is there. It involves lot of fine tuning, fine tuning at you thing. It is easy to parallelize, that is, when you want the value of Y for four different cases, it can be sent to four different computers or four different processes.

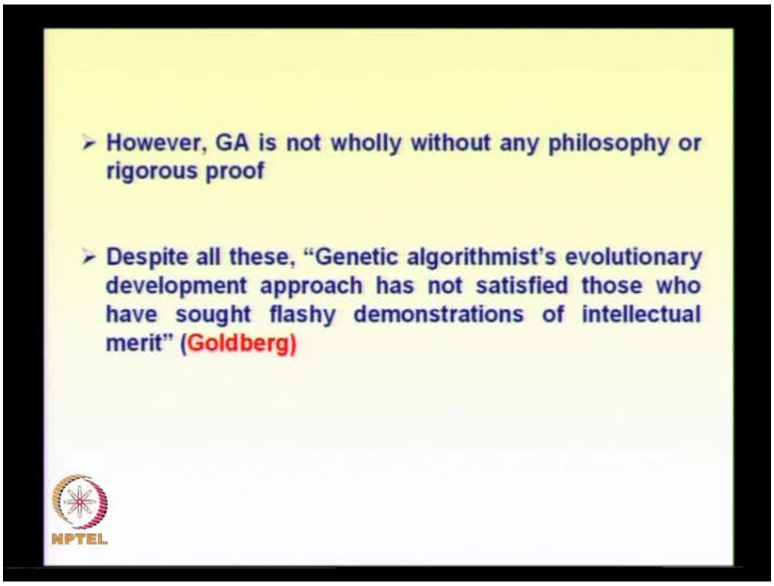
So, it can easily be parallelized, and you can speed up. So, it is easy to get started; today itself you feel that you are an export in GA, it will give that feeling, but once you start working only, it will pain you. So, it is quite difficult to make it work well. Teaching may not be very difficult, but teaching well is very difficult.

(Refer Slide Time: 52:55)



Summary: nature worries, today everybody is present, ok. So, nature worries about what about, about that which works. Nature propagates and procreates only that which survives. She has no time for funda based proofs. So, she has very little, so, she has; so, she has little hope of getting her evolution paper accepted by mathematically rigorous people. She cannot validate or whatever, right; she cannot do any independent study.

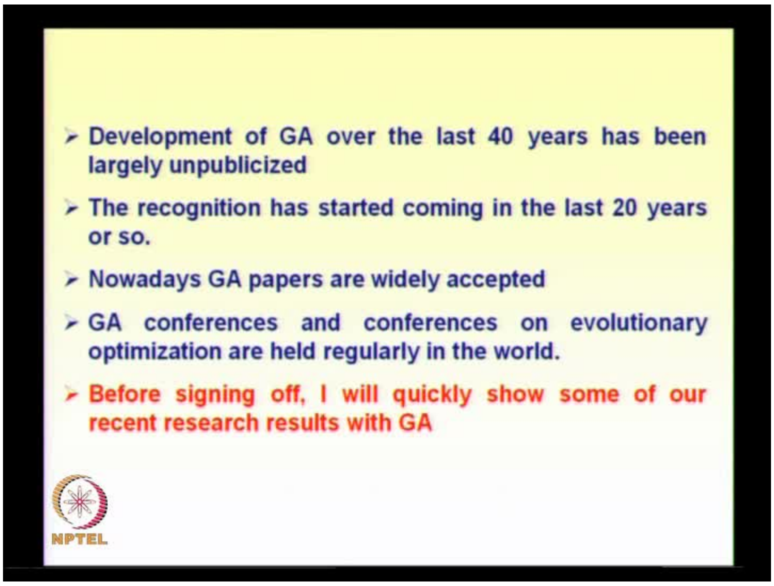
(Refer Slide Time: 53:23)



However, GA is not without any philosophy. Despite all these, mathematicians, pure mathematicians are always treated GA like magic, there is nothing in this and all that.

Some people have tried to look at this (( )) and this thing, and they have tried to, some of these operations they have tried to mathematically prove. But, nowadays, it has been, seen so many problems; it has been successfully applied to so many problem; GA is now widely accepted, there is no problem with.

(Refer Slide Time: 53:51)



Last 20 years, it has been accepted. Now, you have got evolutionary optimization, conferences, GA conferences all over the place. So, on this note, I will, people are saying that the, after 1000 of years, only lot of brain, because anyway people are not moving on. So, the legs, and we lost the tail, right, like that, we will lose other parts, because mostly it will be only, everybody will be sitting in front of the, cell phone will be embedded here. So, pick it up and this thing. So, we will work out a problem in GA, in the next class.