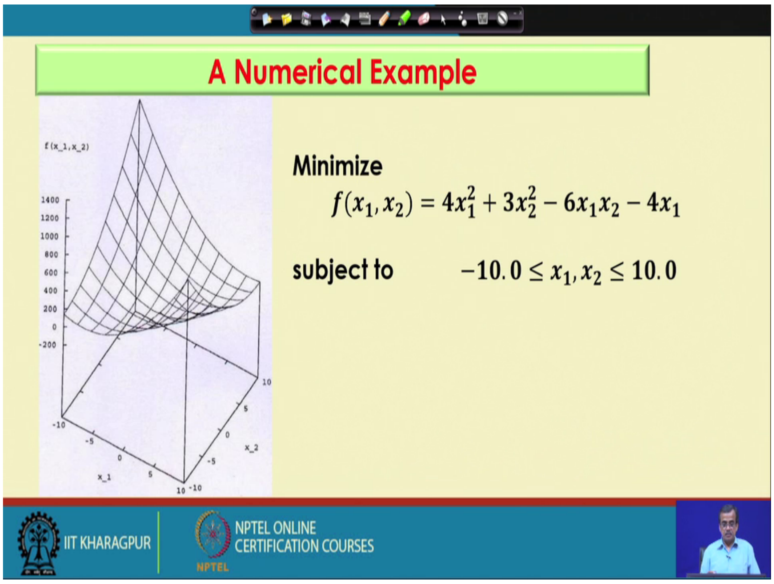
Traditional and Non-Traditional Optimization Tools
Prof. D. K. Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 03
Traditional Methods of Optimization (Contd.)

Let me solve one numerical example using steepest descent method.

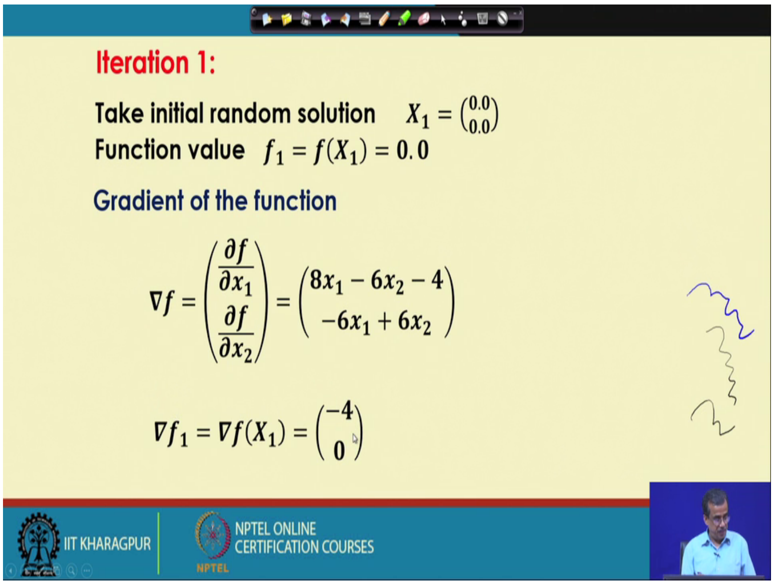
(Refer Slide Time: 00:24)



The problem is as follows. So, I will have to minimize y is a function of 2 variables x one and x 2 and that is equals to 4 x 1 square plus 3 x 2 square minus 6 x 1, x 2 minus 4 x 1 subject to the condition that x 1, x 2 align between minus 10 to plus 10.

So, this is an unconstraint optimization problem and there is no functional constraint here. Now this shows the function plot and here we can we will have to find out the minimum point.

(Refer Slide Time: 01:17)

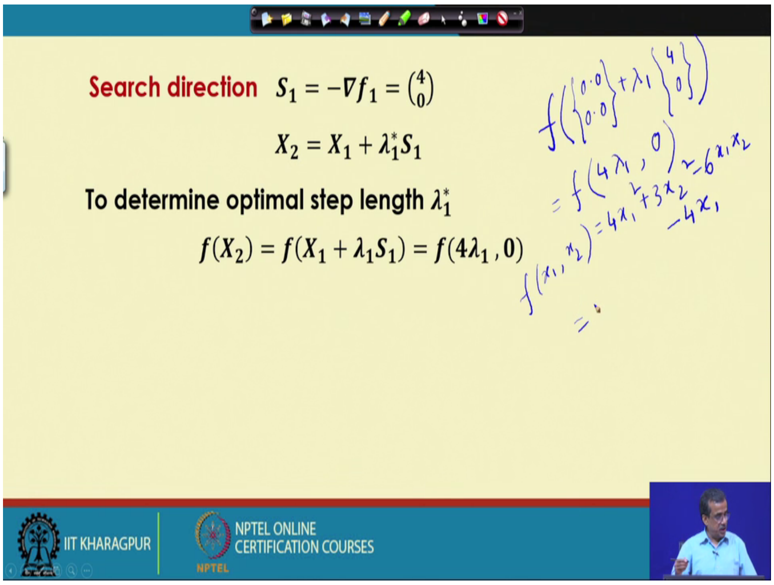


Iteration 1, we take the initial random solution that is capital X 1 that is 0 0, now capital X 1 is a collection of the small x 1 and small x 2 values we calculate the function value that is f 1 and that is nothing, but f of capital X 1, and if you substitute the values of x 1 x 2 in the objective function. So, he will be getting 0.0.

Now, I will try to find out the gradient of this particular the objective function. Now gradient of this particular function is nothing, but delta f is a collection of the partial derivative of f with respect to x 1 partial derivative of f with respect to x 2, and if we try to find out the gradient. So, del f del x 1 that will become 8 x 1 minus 6 x 2 minus 4 and the partial derivative of f with respect to x 2 is nothing, but minus 6 x 1 plus 6 x two.

So, this is the gradient of this particular the objective function. Now on once I have got the gradient now I can find out the value of this particular gradient with respect to capital X 1, and if I substitute the values of a small x 1 and x 2 here. So, I will be getting minus 4 0.

(Refer Slide Time: 03:11)



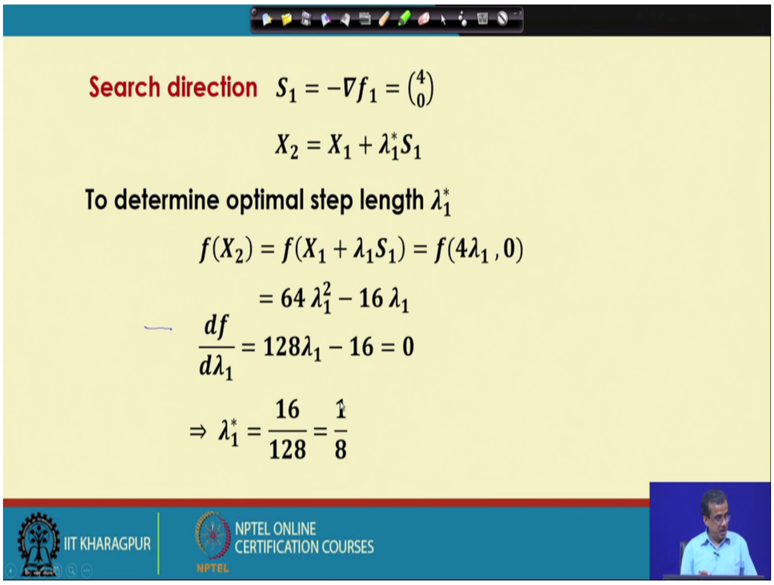
Now, the search direction is nothing, but opposite to the gradient. So, minus delta f 1 and that is nothing, but 4 0, because this is a steepest descent method. So, we will have to move in a direction opposite to the gradient.

Now, x 2 that is a next solution is nothing, but the previous solution x 1 plus lambda 1 star multiplied by S 1. Lambda 1 star is nothing, but the optimal step length and S 1 is the search direction. To determine the optimal value of this step length that is lambda 1 star we follow a particular method. So, we try to find out f of capital X 2 and that is nothing, but f of capital X 1 plus lambda 1 S 1.

Now, if I just substitute the value here. So, this will become like f of x 1; x 1 is nothing, but is your 0 comma 0 and 0 plus lambda 1 and S 1 is nothing, but 4 0. Now this can be written as f of. So, 0 plus 4 lambda 1 that is 4 lambda 1 comma 0 plus 0. So, this will become 0. Now we substitute this in the objective function the expression of the objective function, now expression of the objective function is as follows like f of x 1, x 2 is nothing, but 4 x 1 square plus 3 x 2 square minus 6 x 1, x 2 minus 4 x 1.

Now, if I substitute the values for these x 1 and x 2 like x 1 is 4 lambda 1, and small x 2 is equal to 0. So, I will be getting one expression for this f x 2.

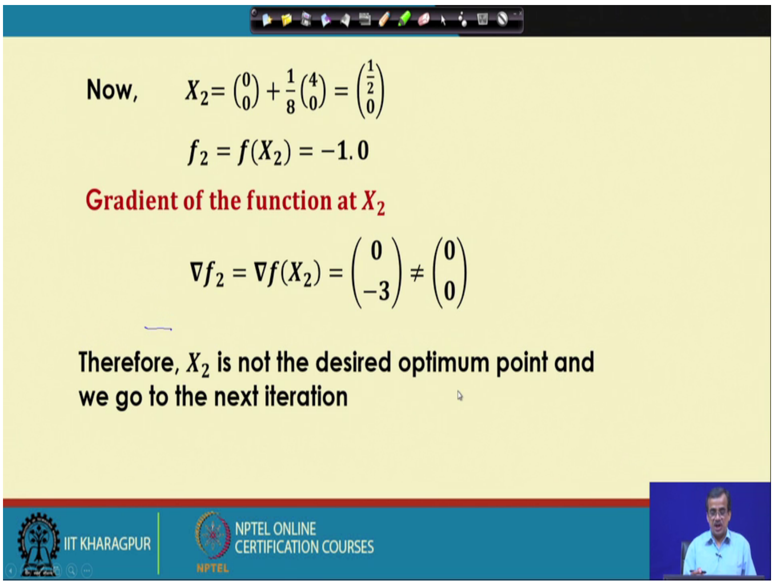
(Refer Slide Time: 05:35)



And that is nothing, but 64 lambda 1 square minus 16 lambda 1. Now here now it has become a function of only one variable that is lambda 1, and to find out the optimal value for this lambda 1. So, what we can do is, we can find out the derivative and put equals to 0.

So, we find out the derivative that is df d lambda 1 and that is nothing, but 128 lambda 1 minus 16 and we put equals to 0, and solve for lambda 1. Now this lambda 1 star this will become 16 divided by 128 and that is nothing, but 1 by 8.

(Refer Slide Time: 06:25)



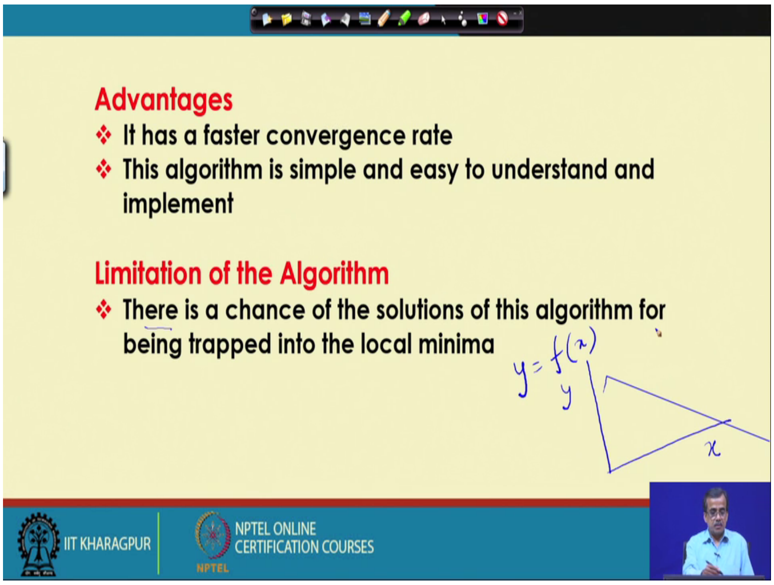
So, this is actually the optimal step length. Now I am going to use this the value of the optimal step length to find out X 2 and X 2 is nothing, but 0 plus lambda star is 1 by 8 into the search direction that is 4 0 and if you simplify I will be getting half 0; that means, small x 1 is equal to half and small x 2 is equal to 0.

Now, f 2 is nothing, but f of capital X 2 and if I substitute the values of small x 1 and small x 2, I will be getting the function value that is nothing, but minus 1.0. Now we try to find out the gradient of the function at capital X 2, now delta f 2 is nothing, but delta f capital X 2 and if I calculate we will be getting 0 minus 3 and that is not equals to 0 0.

So, the termination criterion is not fulfilled. So, therefore, we say that x 2 is no not the desired optimum point and we will have to go for the next iterations. Now this completes one iteration of this particular algorithm, similarly we will have to use a number of iterations and ultimately he will be getting the optimal solution using steepest descent method.

Now, I am just going to see the advantages of these steepest descent method, now this is a very fast algorithm. Now as the search direction is opposite to the gradient and by definition gradient is the direction along which the rate of change of the function will be the maximum. So, this is the in fact, the fastest algorithm.

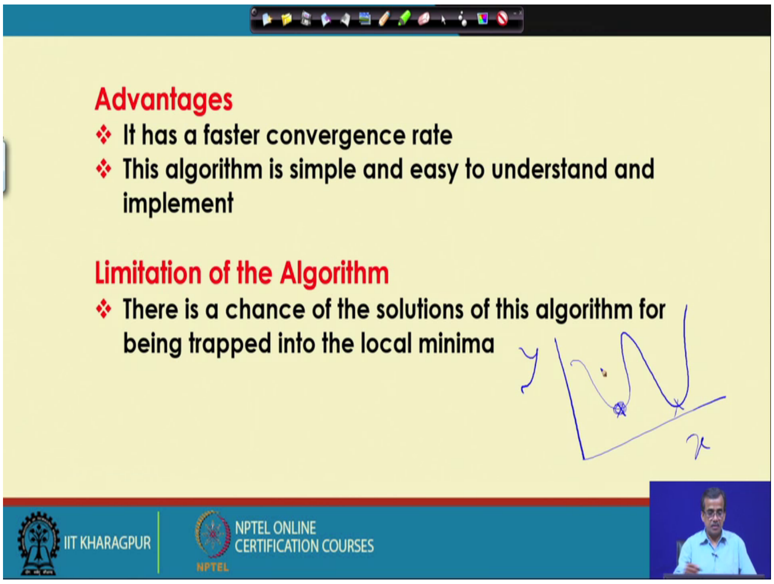
(Refer Slide Time: 08:30)



Now, this algorithm is very easy to implement and to understand and it is very straight forward; however, it has got some limitation which is as follows. Now here in this particular algorithm there is a chance of the solutions for getting trapped into the local minima. The reason is the gradient is a local property. So, there is every possibility that the solution is going to start at the local minima. The concept of local minima let me explain now supposing that I have got a function y is a function of only one variable.

Now, if I just plot it supposing that this is y this is x, and I am getting a plot now this is very hypothetical plot now the plot is as follows.

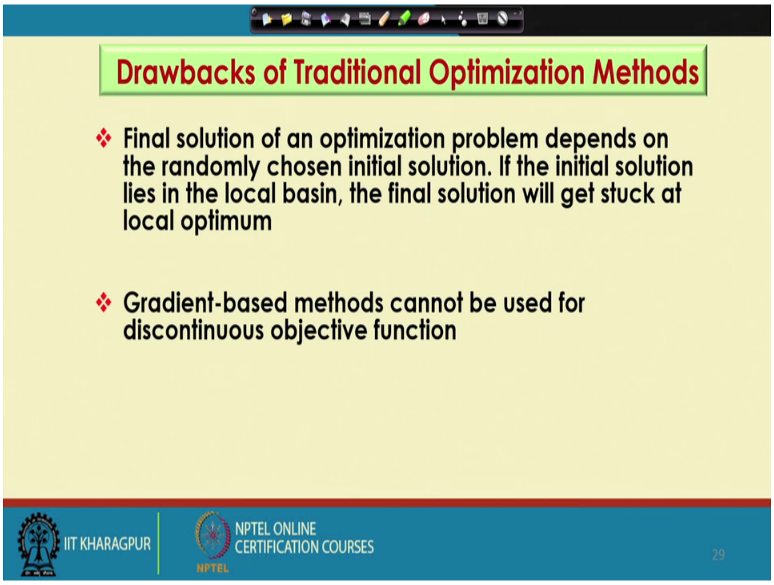
(Refer Slide Time: 09:35)



Now, here, this basin is nothing, but the local basin and this is a global basin and this is a minimization problem, this point is actually the locally minimum point and this is the globally minimum point.

Now, here there is a possibility that if we start from the local basin if the initial solution is lying in the local machine. So, there is a possibility to gradually it is going to hit this locally optimal solution, and as there is no guarantee that the initial solution is going to lie in the global basin, I have got no guarantee that I will be getting the globally minimum solid solution always. So, this is one of the limitation of this particular the algorithm.

(Refer Slide Time: 10:36)



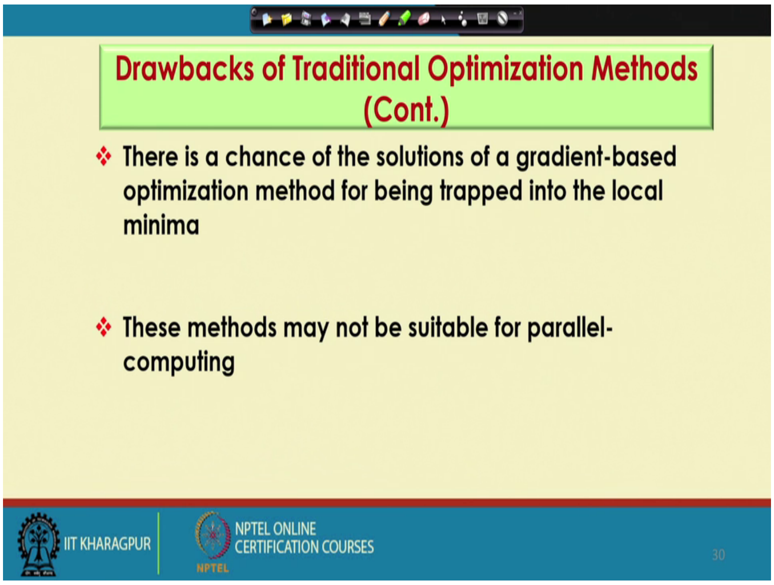
Now, I am just going to discuss the drawbacks of traditional optimization methods. Now before I start with the discussion on the drawbacks, let me tell you that till now we have discussed a few traditional tools for optimization I started with the analytical approach and we concentrated on a problem having only one variable and we have seen how to solve it. So, after that we discussed one numerical method namely the exhaustive search method, and we solve to an numerical example also to explain it is the principle.

So, after that we will concentrated on a direct search method namely the random work method, and we solve another numerical example. And at the end I discuss the working principle of one gradient based method that is steepest descent algorithm and its principal has been explained with the help of one numerical example.

Now, with this I am just going to put the drawbacks of the traditional optimization methods. Now in traditional optimization tools we will start with one initial solution selected at random, and the final solution depends on the quality of the initial solution. Now if you select the initial solution in the local basin. So, we will be getting optimal solution and if you are lucky enough to select the initial solution in the global basin then only you will be getting the globally optimal solution.

Now, the gradient based method like the steepest descent method is not suitable for the objective function, whenever it is found to be discontinuous because we cannot find out the gradient of the objective function.

(Refer Slide Time: 12:49)

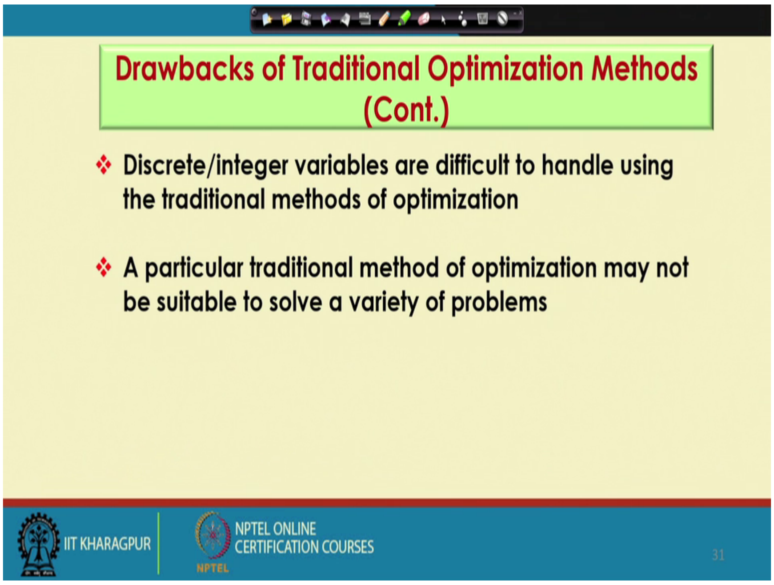


Now, here there is a chance of the solution of the gradient based method to get stuck at the local minima. Now local minima is actually a minima, which is not the globally optimal one and we thought that there is the optimal, but there is a chance of further improvement of the solution, which we could not hit. So, this is an inherent problem of the gradient based method. Now this method the traditional methods for optimization cannot be implemented in parallel computing.

Now, what you do in parallel computing we try to minimize the effective CPU time for a complex optimization problem, because for a complex optimization problem it takes more amount of time to give the solution. So, here what you do is we use a number of CPUs and the minimum number of CPU is 3, and we distribute the total competition in several computers several CPUs and we use one algorithm that is called MPI algorithm that is message passing instruction. Now with the help of this MPI algorithm, the information from one computer is going to be passed to the next computer and vice versa and this multiple CPUs are going to talk to each other I should say and they are going to send information to each other and ultimately we will be getting less competition time the effective less competition time to solve this particular the problem.

Now, as the traditional optimization tool, works based on only one initial solution, we cannot implement the concept of parallel computing, in case of the traditional optimization tool.

(Refer Slide Time: 15:03)



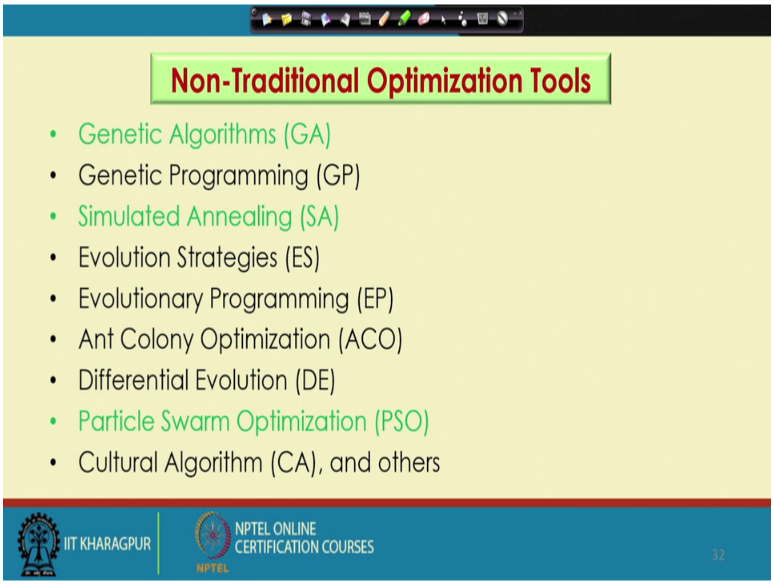
Now, here in traditional optimization tool we face some problem to tackle the optimization whenever there is or there are integer variables and tackle in the integer variables is bit difficult and sometimes we face some problem that is mixed integer problem, and there it creates for the problem and to solve that special type of problem. So, we use a special type of algorithm that is called integer programming technique.

Now, here in traditional tools, a particular traditional optimization tool may not be suitable to solve a variety of problems. So, it is not robust. Now these are the drawbacks of the traditional optimization methods and to overcome these actually what I do we take the help of some sort of nontraditional optimization tools.

Now, before I go for this nontraditional optimization tool, let me summarize whatever I discussed on the traditional tools. We have discussed the working principle of a few traditional optimization tools. Now out of all search tools there is one tool which we have already discussed that is the random work method, in which the search direction is selected at random and there is no guarantee that it is going to hit the optimal solution and it may take a large number of iteration. So, in one side we have got the random work method and on other side we have got a very structured algorithm like the steepest descent algorithm, for the search direction is predefined and that is nothing, but the opposite to the gradient.

And as I told that this is the fastest algorithm. So, on one side we have got the random work method, other side we have got the steepest descent method and each of the methods is having its own merits and demerits. Now what you want is we want some robust optimization tool, which is in between of this; your the random work method and the steepest descent algorithm and it will be robust to solve a variety of complex real world optimization problem, and that is why the concept of nontraditional optimization tool has come into the picture.

(Refer Slide Time: 17:52)



Now, I am just going to discuss little bit on nontraditional optimization tools. Now this nontraditional optimization tools or working based on the principle our nature is going to follow. So, these are all nature inspired tools or biological inspired tools. Now if I see the literature we have got a large number of nontraditional tools for optimization the list is very big. For example, say we have got the technique like genetic algorithm, which was proposed in the year 1965, then we have got genetic programming then simulated annealing, we have got evolutionary strategies, then we have got evolutionary programming ant colony optimization, differential evolution that is the modified version of genetic algorithm, then we have got particle swarm optimization, we have got cultural algorithm this is once again another modified version of genetic algorithms and others. The list is very big we have got some other nontraditional optimization tools also.

Now, here in this particular course what I am going to discuss some of the nontraditional optimization tools and I will be discussing their working principles with the help of some suitable numerical examples. For example, I am going to start with the working principle of genetic algorithm, now if you see the literature we have got various versions of this particular genetic algorithm, it includes the binary coded genetic algorithm, the real coded genetic algorithm, grey coded genetic algorithm, then comes your the micro g a messy GA and so on. Now here in this particular course I am going to discuss in details the working principle of the binary coded genetic algorithm, the real coded genetic algorithm, the first genetic algorithm like the micro GA, visualized interactive GA, and then I am going to discuss a special type of GA for solving the scheduling problem and that is known as the scheduling GA. I will also be discussing the principle of simulated annealing.

Now, here actually what we do is, we try to model artificially the cooling process of molten metal and try to solve the minimization problem. And I am also going to discuss in details the working principle of particle swarm optimization, and here actually we are having a few advantages over the genetic algorithm. So, those things I am going to discuss in details with the help of some numerical examples.

So, all such tools the nontraditional optimization tools are having their own merits and demerits, and we will try to take their merits and we can try to overcome their demerits and we can also develop some sort of the intelligent optimization tools. So, gradually I am just going to move towards that particular the direction.