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

Lecture – 27
Intelligent Optimization Tool

I am going to start with another topic and it is on the concept of intelligent optimization tool. Now, if you see the real-world optimization problems, these problems are complex in nature and it is it is dynamic also. And moreover, there could be some uncertainty in the problem definition and there could be a possibility some of the design variables may change. We may not get a very well defined the expression for the objective function and there could be some discontinuity in the objective function.

But, our aim is to get some optimal, the near optimal or globally optimal solution within a fraction of a second. Now, to get this particular online solution, for this type of difficult problems, so, we will have to design and develop some more efficient algorithm like intelligent optimization tool. Now, if you see, in any optimization algorithm, there are 2 things which elliptic control. These are the search direction and the step length.

So, these algorithms are, in general, iterative in nature; that means, at each of the iteration, so, we will have to find out what should be the optimal step length and what should be the optimal search direction. Now, whenever we use any such optimization algorithm, to solve the optimization problem, we try to get a good accuracy in the solution and at the same time, the search speed should be good enough.

So, we want to get accurate solution, but at the cost of no computation. And if you want to implement the concept of this particular optimization or if you want to use the optimal solution online, so, this particular optimization algorithm should be able to provide the solution within a fraction of second. Now, keeping all such things in mind, we thought about the concept of this intelligent optimization tool.

And another thing I want to mention, that our experience says that, a particular tool is able to solve only a particular type of problem and the performance of these particular optimization algorithm our problem dependent. But what do you want is, to find out to design and develop some optimization tool which is function independent. So, with

keeping those things in mind, we thought about the concept of the intelligent optimization tool.

(Refer Slide Time: 03:35)
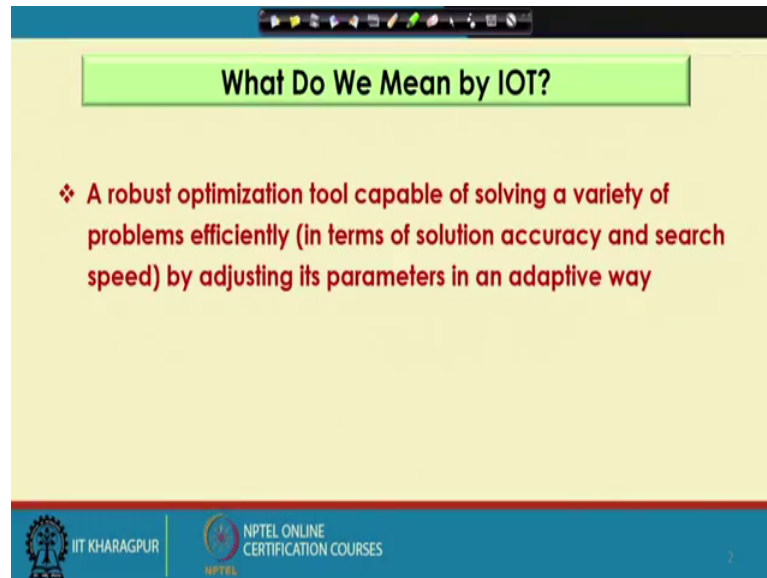
Now, let us see what do you mean by the intelligent optimization tool. So, by intelligent optimization tool, we mean that, this particular tool has to be this particular tool has to be robust. Now, robust in the sense, that it should be able to solve a variety of problem efficiently in terms of the solution accuracy and the search speed.

And how to get it? So, through a large number of iteration. So, this particular tool will have to adjust it is parameters in the adaptive way. So, by intelligent optimization tool, we mean, it is a robust optimization tool capable of solving different types of problems efficiently by adjusting in parameters in an adaptive way.

So, this is actually the concept of the intelligent optimization tool. Now, that means, how to inject some sort of intelligence inside the optimization tool, so that, it can take the decision as the situation demands. Now, let us see, how to implement. But before that, let me try to fill the need for this intelligent optimization tool which I have already discussed little bit.

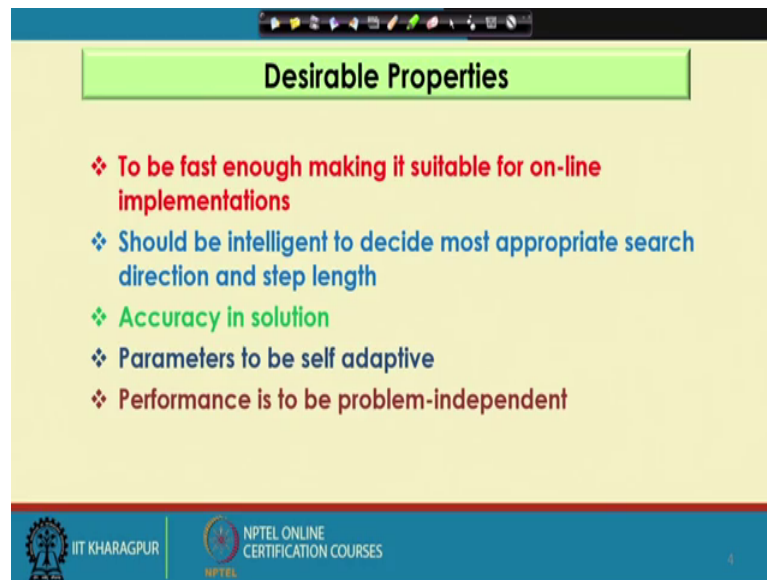But let me discuss point wise.

(Refer Slide Time: 05:05)

Now, this real-life optimization problem as I told, are very difficult to solve because, it has got a few problems like your we may not get a very well defined objective function. There could be discontinuity in the objective function; it could be dynamic in nature and all such things. The condition of the problems change frequently and there could be uncertainty also.

Now, our aim is to find online optimal solution of this particular the complex optimization problem and that is why; we will have to take the help of intelligent optimization tool.

(Refer Slide Time: 05:55)

Now, the desirable properties like how can you declare that a particular optimization tool is intelligent.

Now, to declare certain properties are to be followed are to be maintained by that particular optimization tool these are as follows; that means, it has to be very fast, so that we can implement online. So, now, for online implementation, within a fraction of second, it should be able to provide that particular optimal solution and this algorithm should be intelligent enough just to decide what should be the step length and what should be the search direction at each of the iteration.

And of course, it will up to maintain that particular the level of accuracy. The solutions are to be accurate at the same time, the alga that this particular tool should be able to provide that particular solution as quickly as possible. And to do that, actually the parameters of this particular algorithm should be self-adaptive; that means, it will have to adjust it is parameters depending on this particular the the situation of the problem.

And, as I told, this particular the performance of this particular algorithm has to be problem it depended. Now, if I see the reason behind switching over from the traditional to non-traditional tools, there are many reasons and out of all the reasons, the one of the most important reasons were actually we should get a robust optimization tool which can tackle a variety of problems.

But unfortunately, we saw the performance of the traditional tool for optimization is problem dependent. And now, it is actually, we use a large number of non-traditional optimization tools and once again we are facing the same problem. the problem is as follows nowadays. We have around 30 non-traditional tools for optimization and this particular number is going to increase like anything.

Now, this was actually not our original agenda of switching over to the non-traditional tool for optimization. The reason why we went for the non-traditional tool for optimization was, how to get one a robust optimization tool which will be able to handle a variety of problems, but as on today, we are again facing the same problem. We have got a large number of non-traditional tools and the number is going to increase day by day.

But once again, we are forgetting the basic need why we switched over to not additional tools for optimization; that is the the robustness. And we have seen that, the moments of these particular the non-traditional tools for optimization is found to be problem dependent. And that is why, we want only one intelligent optimization tool, say, non-traditional optimization tool, which will have all the good properties and which will be intelligent enough to tackle a variety of problems with equal ease and efficiency to find out the globally optimal solution.

So, our aim is to develop one optimization tool which will be problem independent; that means, the same intelligent optimization tool will be able to solve a variety of optimization problem by adjusting it is parameters in an adaptive way. That is actually the concept of the intelligent optimization tool. Now, how to achieve it? Now, to achieve this particular intelligent optimization tool, you will have to follow a few steps.

(Refer Slide Time: 10:18)

Now, what you will have to do is, we will have to study the topology of the objective function. Now, as I told that, supposing that I have got a function 10 variables, y is a function of 10 variables and this particular function, I will have to say optimize say maximize or minimize, the surface of this objective function is in 11th dimension which we cannot visualize. So, for the purpose of visualization, we will have to transfer this particular data to lower dimension say either 2 dimension or 3 dimension for the purpose of visualization.

And once we are able to visualize, the surface of the objective function, will be able to decide what should be the most appropriate search direction and what should be the optimal value for this particular the step length. Now, if we want to design and develop the intelligent optimization tool, so what we will have to do is, we will have to incorporate some level of intelligence to the algorithm, so that, it can find out some optimal step length and search direction by carrying out a search or the the surface of it is objective function.

So, this is actually one of the way and one of the steps to be followed if we want to design and develop the intelligent optimization tool. Next is, the solution should have memory; that means, whatever happened in the previous iteration or previous to the previous iteration, that ought to be memorized. And using this particular past record, that is the history, so the future direction will be decided.

So, to use this particular past record for the history, the all the solutions are to be memorized. Now, to memorize the solution, definitely it will take a little bit of space, but we will have to work more on this how to memorize this particular solution in a better way, so that, it can consume less space to memorize that particular the data. This is one area where, we will have to work more extensively like how to memorize this solution by using less space.

This is one of the research areas in this particular the field. Now, we should be able to provide the feedback system in this particular optimization tool, so that, it can decide it is parameters in a very adaptive way as the situation demands. Now, let me take one example. Now, if I want to use a genetic algorithm, to solve one very complicated real-life optimization problem, so, what will have to do is, it is parameters like your the probability of crossover, the probability of mutation, population size and so on, those can be adjusted in an adaptive way, during this particular your the GA search and this particular intelligent optimization tool should have this particular the provision.

(Refer Slide Time: 14:06)

Now, how to achieve it? To achieve this, actually a lot of work is going on. Many people are trying in different directions.

Now, I am just going to tell a few methods the way these particular things can be implemented. A few things I have already discussed like, we will have to see the

topology of the objective function just to find out what should be the search direction what should be the step length.

Next is, you will have to find out what should be the optimal values of the parameters. And next is, we have to design and develop a suitable restart strategy. Now, supposing that, I am running a genetic algorithm for say, 1000 generation to solve a very complicated optimization problem, supposing that, up to say 20 generation or 15 generation, I am not getting any significant improvement in the values of this particular optimal solution.

So, might be from 15 to 20 or 15 to 30 I did not get any improvement. Now, this particular the problem could be due to the premature convergence or it could be some sort of local minima. Now, if it is premature convergence, now, unnecessarily we should not allow this particular GA to run up to 1000 iterations or 1000 generation. Instead, what we can do is, we can stop the GA and we can restart GA and by doing that. So, the problem which we are facing that is, the local minima problem or some sort of premature convergence, we can avoid and we can restart this particular algorithm and we can help this particular GA to find out what should be the globally optimal solution as quickly as possible.

Now, if you see the literature, now there are several restart strategies available. Now this is another a very good area where a lot of work is going on and many people are working like how to how to design and develop a most suitable or efficient restart strategy for this particular the genetic algorithm. The next point is efficient operators. So, we will have to design and develop the efficient crossover operator's efficient mutation operators to carry out this particular search in a very adaptive way.

Now, till now, we have discussed a few modules or few stages how to make it intelligent now, if you want to design and develop an ideal intelligent optimization tool, so all such the steps are all such stages which I discussed are to be put in one complete module of optimization tool. Now, the the the the the the parent algorithm, it could be genetic algorithm, it could be particle swarm optimization or any other algorithm, but all such extra features, so, you will have to incorporate to this the parent algorithm like genetic algorithm or say particle swarm intelligence and got the particle swarm optimization algorithm, just to make it more intelligent, so that, we can tackle very complex real

world problems in a very efficient way and we can make it faster and we can get this particular very efficient solution.

Thank you.