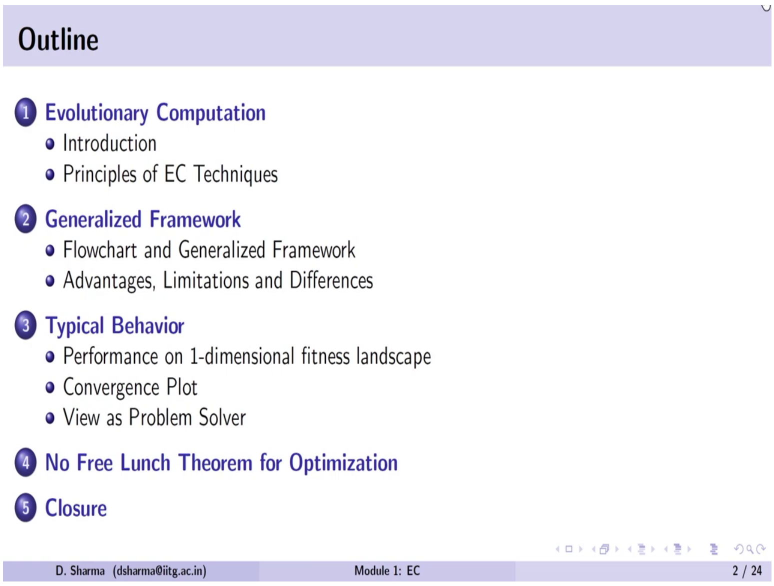
Evolutionary Computation for Single and Multi-Objective Optimization

Dr. Deepak Sharma
Department of Mechanical Engineering
Indian Institute of Technology, Guwahati

Module - 01
Lecture – 02
Introduction to Evolutionary Computation

Welcome to the session 2 of module 1 that is Introduction to Evolutionary Computation.

(Refer Slide Time: 00:38)



In this particular session I will be covering the Evolutionary Computation, introducing it and thereafter we will discuss the principle of EC techniques. Now, you will realize that in the principle there are different ways we can have those evolutionary computation techniques. So, those principles we will be discussing in EC techniques. Thereafter we will be discussing about the generalized framework.

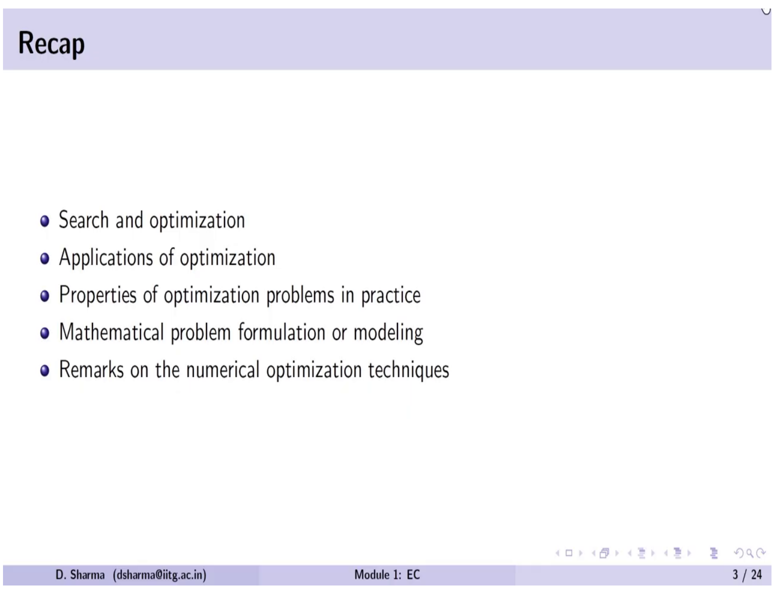
So, the motive is when we are going to discuss the generalized framework, so that framework will be applicable to many EC techniques. So, in that case we will first discuss the flow chart and in that flow chart we will discuss each and every steps that are needed in EC techniques.

Thereafter, we will discuss the algorithmic way of presenting the EC techniques. Afterwards we will talk about the advantages, limitations and differences. Now, these advantages are important because we will know that in what situations EC techniques are going to help us to solve large and complex problems.

Similarly, we should also know the limitations before applying EC techniques for any particular problem. In the differences, we will see how these EC techniques are different from numerical optimization techniques. Thereafter we will discuss about the typical behavior mainly we will see the behavior on a 1-dimensional fitness landscape. So, we will see how EC techniques will perform on them. Similarly, the convergence plot we will see.

Then there is a Goldberg’s view for as a problem solver for EC that we will discuss in detail. And finally, there is a no free lunch theorem for optimization that is important in this context why because we have to realize that a one method can solve our large variety of a problem, but cannot solve the whole or the all kind of optimization problems and finally, I will close this module 1.

(Refer Slide Time: 03:00)



So, before we go let us have a recap. So, in the last session we discussed about the search and optimization. I took an example where I showed you that we start with the point randomly in the search domain and then we move in a direction where the objective function will improve. This will give me an another point.

So, that point from that new point we again find a direction, so that we can move it and find the optimum solution. So, in this way we are keep on iterating, we are keep on searching in the direction, so that finally, we should converge to the optimum solution. So, therefore, in an optimization it involves search and optimization.

So, we are looking for a set of decision variables that will be minimizing our objective function or maximizing our objective function subjected to various constraints. Thereafter we talk about the application. So, the whole motive of showing different kind of application in different domain is that GA has been used or other evolutionary computation techniques have been used for solving large and complex problem.

We started with car design problem where computer models were made GA was used to find the optimal design of the racing car. In the in this particular analysis what we realize that instead of putting lot of effort money in doing so many of experiments and coming up with an optimum solution, the computer models can give you desirable solution for the required problem.

Similarly, the engineering design problem as we know in every area the engineering design problems are important. It is because we always look for the solutions or we are always looking for the optimal design of our structure. So, therefore, our aim is to optimize the structural and operational performance of those structures. I showed you various examples from turbines to the heat exchanger to the mechanical component and several component. So, what you will realize that it is another computer aided engineering design problem.

Thereafter, we had one of the interesting and quite old problem that is on travelling salesman problem. It has a wide application. So, since the problem is little complex. So, in this particular say in this particular scenario evolutionary computation techniques will help us to find an desirable an optimum solutions.

Now, it is important to note that the when the number of nodes are a small, then some techniques can be used to solve it. However, when we are going to solve a real word problem which is in generally large and complex there EC techniques will help us to solve such kind of a problem.

Thereafter, we discussed about telecommunication problem that has been a study which was done in one of the cities and they found that if they can optimally place those tower cells, then everyone is going to get the signal. So, in that case they were able to reduce the overall project cost meaning where to place those towers as well as every user was getting the desired mobile network for their applications.

Similarly, we also talked about data mining. Now, in these days data mining is very important for us why because we have lot of data whether it is related to a machine or it is related to some process. So, looking at the data we can extract lot of information. Now, optimization is inherent part of a data mining. We started with regression where we want to have a best fit curve. We what we can see here is the optimization can be used, so that we can get a desired solution or desired accuracy for prediction, for parametric optimization etcetera.

Thereafter, we talked about classification, mainly the binary classification we discussed. In that one we can divide the data into two different parts or classify them into two parts, so that we can say whether the particular machine is working fine or a not or if a person has a disease or a not. Thereafter we discussed about clustering, a clustering of different kind of a data has an application in say image processing and pattern recognition.

Similarly, we discussed other examples which are important for us and then there was a long list of examples which is which shows there is a scope of an optimization and as and when we discuss the practical problem we need an optimizer which can solve large and complex problem. Then we discussed about the properties. Now, these properties are important because those properties will tell the complexity of an optimization problem.

So, therefore, we discussed one by one from a non-differentiable function, we started with a non-differentiable function; thereafter we discussed the discontinuous function. All these kind of problems which has difficulties different difficulties level that pose challenge to an optimization problem to find an optimum solution. We also discussed about the discrete search space and that is also a very difficult problem when we are going to solve them.

Afterwards we also talk about robust and reliable solution. Now, in many scenarios we are looking for the robust solutions. So, when we optimize any function it will eventually give you a solution which is at the peak. However, at that peak, small change in the value of the variable will drop the objective function largely. So, in that case such kind of a optimum solution may not be desirable. So, therefore, we look for those reasons where the optimum solution is should not change much with a change in the decision variable.

Similarly, we had reliable solutions. Now, reliable solutions are important when we have variables which are which have the uncertainties. So, the deterministic optimization solution will become infeasible for us. Therefore, we look for a reliable solutions and that makes the problem complex which can be easily solved by evolutionary computation technique.

One more problem characteristics we discussed that was multi modal optimization. Now, multi modal in a sense that we have multiple point at which our objective function is maximum. So, in that case what we will realize that when we are going to solve it we may end up getting just one solution.

However, since, this particular solution we can use for designing some product or it can be a strategic decision in an in our organization, so, we would like to see all those solutions, compare them and choose the best. So, therefore, the objective is to find all the solutions for a multi modal problem and that becomes a challenge for EC techniques and other optimization formulation.

After understanding the different kind of properties of optimization algorithm we move to the mathematical problem formulation. So, in general any kind of a problem if we want to write in an in a mathematical way then we have to write in terms of objective function in terms of constraints. We had two types of constraint followed by the variable bounds. So, everything includes a one problem formulation.

Now, a afterwards we discussed about the decision variables so, how we can find the decision? So, as we know that there are different parameters that are involved with the decision involved with the problem optimization problem. So, what we do here is we list each and every parameter. Then with the help of sensitivity we can identify that what are those parameters which are sensitive to the problem.

So, for example, if there is a small change in those parameters there is a large change in the objective function. So, those parameters will become a variable or a decision variable for the given problem. So, accordingly we decide decision variables and rest of the parameters we keep them constant, thereafter we decide what type of parameter it is or decision variable it is it can be binary, it can be discrete or real or permutations.

Thereafter, we discuss about the thumb rule. So, the first thumb rule was based on the number of decision variables. It is because the number should be less so that our optimization algorithm will be effective and accurate to find an optimum solutions. Thereafter, we talk about the constraints. So, as we know the constraints are nothing, but the restriction on certain phenomena or a process.

So, when we design our constraints so, we look what kind of constraint we can have in our problem. So, we can have inequality constraint so, where we have a sign for example, greater than type or smaller than type. So, these are the constraint which are very common in most of the engineering optimization problems.

Thereafter we had equality constraints. Now, those equality constraints are also these are very rigid constraints why because they have a right hand side equals to the left hand side that makes the problem complex to solve it. So, generally it is advisable that a one equality constraint can be converted into two inequality constraint.

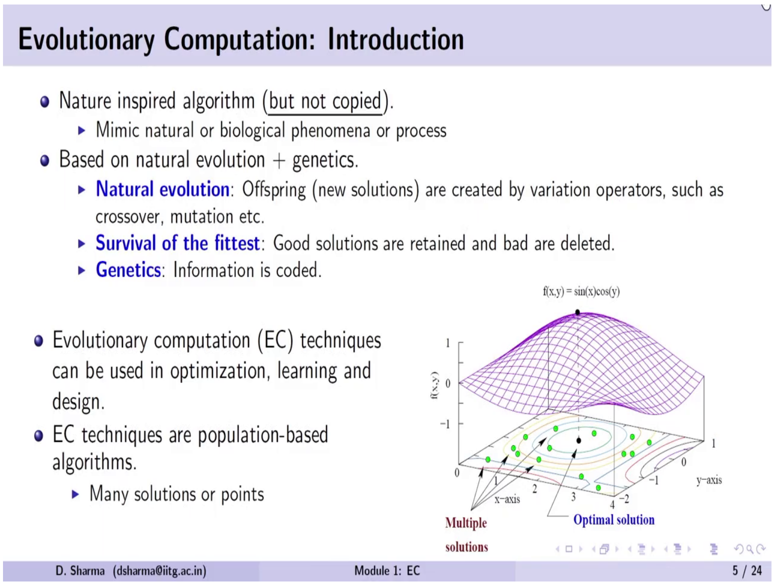
So, based on that the second thumb rule that was decided that was made is that we have to keep the number of complex equality constraint less in our problem formulation. Then we move to the objective function. So, as per our requirement either we will be minimizing the function or maximizing the function. Now, what we realize that whenever we are working or we are using any optimization algorithm they are generally made for minimization or maximization task.

So, suppose if the algorithm which we have that is for minimization. So, in that case if I want to maximize it so, we can use the; we can use the duality and that duality will easily convert our maximization problem into the minimization and a vice versa and thereafter we have to set up the variable bounds. After having the overall idea about the problem, formulation the properties of practical optimization problems and application, then we move to the remarks on the numerical optimization technique.

Now, these remarks are important why because we have understood from those remarks that the algorithms or the techniques which are based on numerical techniques may be efficient for certain class, but not for the other class. So, that is making small set of problems that can be solved by such kind of a techniques. Even if the constraints are there or there is a; there is a uncertainty in the variables, all these problems poses lot of challenges to the numerical optimization techniques.

So, that everything put together require a one particular algorithm that should be flexible enough to deal with such kind of a challenges. So, the answer to this question is evolutionary computation techniques. With this recap let us move to the evolutionary computation.

(Refer Slide Time: 15:59)



Now, first of all what is evolutionary computation techniques? We have to understand that evolutionary computation techniques are nature inspired algorithm. Nature inspired means that whatever we see in our natural phenomena or biological process we mimic them.

So, when we mimic them we try to understand their phenomena we come up with the operators. Those operators will help us to generate new solutions as well as take us to the optimum solution or near to the optimum solution. Here it is important to note that we say that; we say that the we mimic the nature or biological phenomena or a process, but we do not copy.

Because generally these process are complex and exactly copying them may not be so useful. However, we have to stimulate that process with the help of operators. So, that it will help us to solve our complex optimization problems. As we as these evolutionary computation techniques, these techniques are based on natural evolution plus genetics.

Now, here the natural evolution means that we are creating new solutions. So, let us look at the biological phenomena. So, if we take a take an example of human being where we need four support and eventually generation by generation now we need only two support to move and we can do so many task which we which was not possible earlier.

So, what is happening is that the new solutions were created by biological phenomena for example, cross over and a mutation. Now, if we look them we get inspired from such kind of a processes and then we make operators that will be mimicking such kind of biological phenomena. So, the naturally evolution says that the new solutions are created by the variation operators. Thereafter, we have a survival of the fittest. Now, good solutions are retained and bad solutions are deleted.

So, survival of the fittest we know that and in this particular theory we say that the solutions which are good that should be emphasized. It is only because these good solutions says that the solutions which are good means they are closer to the optimum solutions. If they are closer to the optimum solution they should be emphasized at each and every stage, so that generation by generation we should reach to the optimum solution.

However, those bad solution should be deleted because they are of no use and they cannot take us to the optimum solutions. Thereafter, we have genetics means the information should be coded. So, the information about the variables that should be coded, so that we can find what is the objective function, what is the constraint.

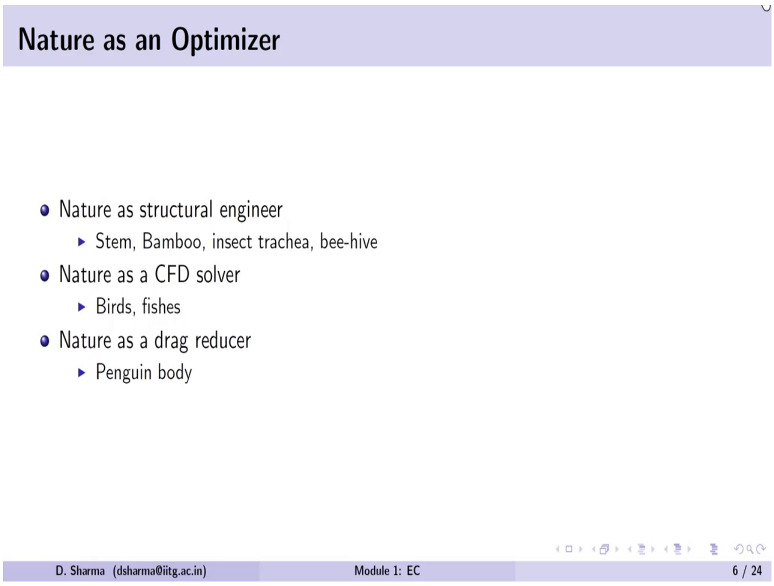
So, looking at these three principles which involve natural evolution, survival of the fittest and genetics, all three put together makes an evolutionary computation technique. Here it is important to note that this - EC techniques can be used in optimization, learning, as well as in the design.

Now, what you will realize that EC techniques are population based algorithm; population word suggest that we work with the set of solution. If we look at the figure on the right hand side now we I have taken the same example as we have taken in the session 1.

Now, the same example where we want to maximize the function sin x and sin x multiplied by cos y. Now, on the x and the y plane you can see there are multiple solutions; now those green solutions where we start. So, again EC techniques generally start with the random solutions. So, these green solutions are randomly generated in the variable space which is constituted with the help of x and a y axis.

These solutions when they are distributed with the help of natural evolution, survival of the fittest and genetics, these solution will move slowly and slowly closer to the optimum solution. So, here so, that is why EC cannot EC computation techniques are called as population based techniques because we always work with multiple points or a multiple solutions.

(Refer Slide Time: 20:53)



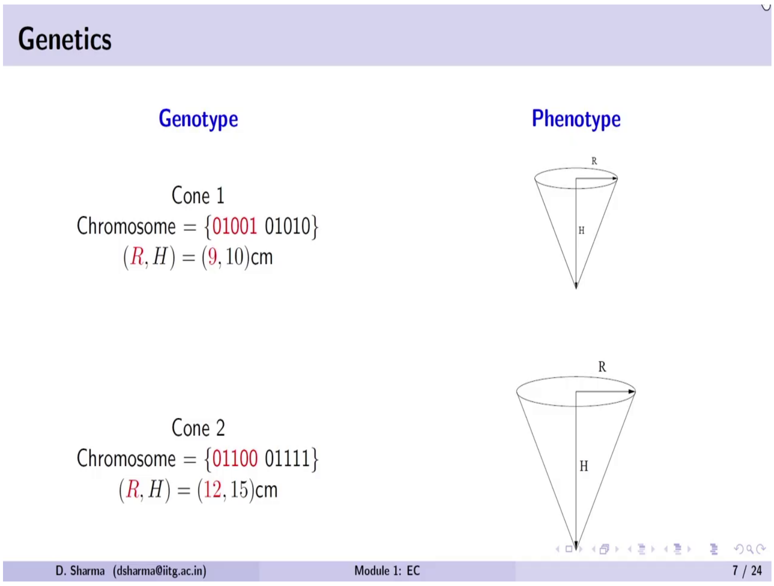
Now, what you can realize in the nature that nature is the best optimizer. If you look at the structure of the stem, bamboo, insect trachea or a bee-hive all of them are very good example if you consider nature as a structural engineer. It is because if you look at the structure of bee hive so, the hexagonal so, regular hexagonal shape that can be used and we know that it has a good strength.

So, this particular structure or the shape of the structure have been used in the practice. So, similarly if you look the structure of these examples you will realize that nature has already been an optimizer and these are the real example where we can get the best optimal structures.

Similarly, if we look at the birds and fishes they fly either in the air or in the water. Now, their body is designed in such a way that there will be a minimum drag they can run fast and similarly other things. So, what you will realize that the birds and fishes are made in such a way that they are the best example of an optimization. Similarly, if you look at the penguin body, so, this is the best example which can reduce the drag. They can swim in the water they can walk on the surfaces. So, their body is made in such a way that it reduces the drag in the water.

So, there are so many examples and that is why we have different EC techniques available whether it is genetic algorithm or particle swarm optimization or it can be cuckoo search, artificial bee colony, ant colony optimization. So, that is why we have understood the process and then we come up with the new kind of algorithm because we know the nature is the best optimizer for us.

(Refer Slide Time: 22:53)



Coming to the those three principles one by one. Let us start with genetics. Now, here the two words are written genotype and phenotype. What we mean by genotype? As we look our human body our information is coded in the gene. So, all that information that is coded we know that this information will tell what should be my eyes the color of my eyes, the ears, height, skin color etcetera.

So, these information is already coded inside it, but when you have those information how you look like physically that is called phenotype. So, the genotype is the information which is coded internally and the phenotype; so, pheno means physically how it looks like.

So, let us take an example here. So, we have taken example of a say cone. Now, as we know the cone can be defined by the radius and the height. Now, here for a cone 1, the internal information is stored in the chromosome. Now, in this chromosome we have two parts; one is the red color, another is in the in the black color. So, to make a difference between the radius and a height I mentioned them in a different color coding.

Now, let us look at the red color. So, this five bits in the red color will be representing what is the radius for the cone. Similarly, the other five bits will be representing the height of the cone. Now, if we decode this binary string in the red color we will get a value of a 9. Similarly, if we decode the black color strings we will get the value of 10.

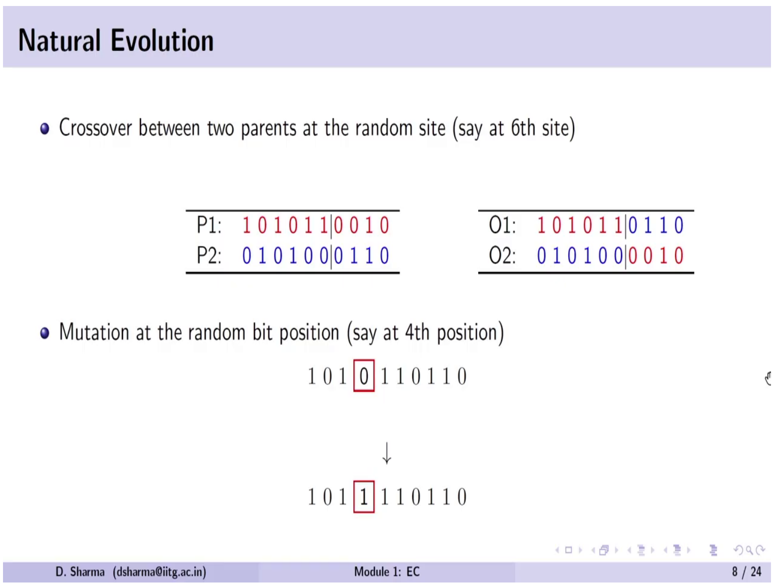
So, if when we go we will get to know the value of R and H as 9 and 10 centimeters. So, we can see on the right hand side how the cone will look like. Now, what you will realize that? If we are going to change internally the chromosome string is it going to change physically. So, here we have another example of cone 2.

Now, look at the green look at the red part of the binary string. Now, here the again the red part represents the radius and the black part of the binary string represents the height. Now, what we have done? We have changed the binary string. Now, you can compare cone 1 and a cone 2.

So, the red part is also changed as well as the blue black part is also changed. When we decode the red part so, here you will realize that the radius will become 12. Similarly, when we are going to decode the black part so, the height will become 15 centimeter.

So, what you will realize that? Internally when the information is changed which is coded in terms of binary string, physically which you can see on the right hand side the structure of the cone or the shape of the cone is changed and that is why genetics stores the information internally and phenotype tells us how it will be look like.

(Refer Slide Time: 26:03)



Now, coming to the second principle which is natural evolution as natural evolution suggests that the new solutions are created using variation operators. So, I have taken example of say crossover operator which is simple and easy to understand at this stage.

So, let us suppose we have two parents P1 and P2. So, the P stands for parent. So, P1 will become parent 1, P2 will become parent 2. Now, here we have chosen randomly the 6th site. Now, let us look what is 6th site. So, this is site 1, this is site 2, this is site 3, this is 4, this is 5 and finally, we get 6.

Now, you can realize that at the 6th site we have drawn a vertical line here. Now, this vertical line is useful because we want to perform crossover. How it works? Now, here in the right hand side you will see we have written O1 and O2; O stands for offspring. So, offspring 1 and offspring 2.

Now, what we have done? I have used the color coding again. So, P1 is represented in red color and P2 is represented in blue color. So, one of the simplest crossover operator is the single point crossover operator. So, this is I am showing you an example that when we chose the 6th site so, from that vertical line we swap the tail.

if we swap the tail now look at O1. So, we have few in red color that is up to 6 and rest of the four are in blue color. Similarly, for O2 you have first 6 bits in a blue color and gethen rest of the four in the red color. So, what you can understand that if we are going to perform crossover like that either the radius or the height of the cone will change. It means that if we are changing the information internally this is going to change our phenotype. So, the cone the shape of the cone will change.

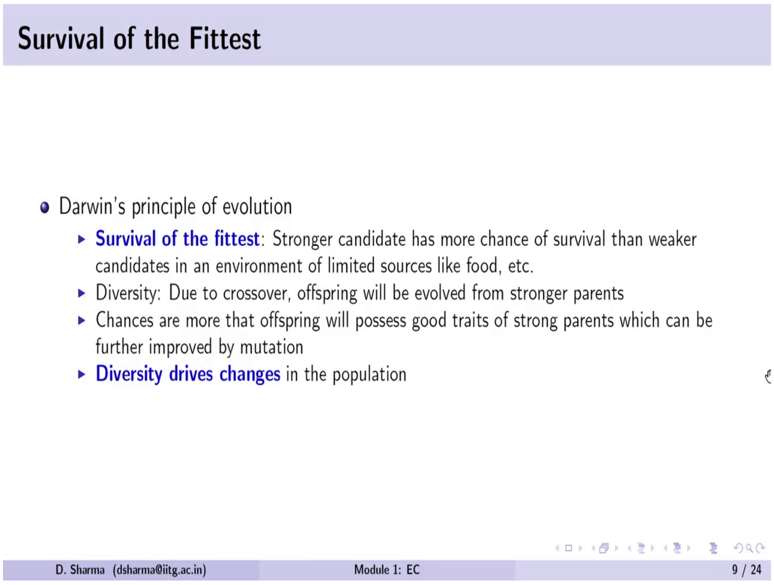
Now, come to the another variation operator which is called mutation. As an example we have taken bitwise mutation. Now, in this particular mutation we generally take one particular position. So, randomly we take at the taken the 4th position. So, if we count from the first 1, 2, 3 and a 4. So, at the 4th position we randomly chosen randomly chosen this particular bit.

So, once we chosen it in the mutation what we do is, so, currently it is written 0. So, what we will do is we will convert this 0 in to 1. Here it is important to note that if suppose we chose 1, then 1 will be converted into 0.

So, currently 0 has been converted into 1. Now, what you will realize that the change in the binary string happened at one of the bit; it means that either the radius or the height of the cone can change. So, this means that the overall shape of the cone will change when we perform the mutation.

So, that is the overall objective in the natural evolution that when we perform variation operator these decision variables or design variables should change when we perform either crossover or an mutation.

(Refer Slide Time: 29:44)



Now, coming to the survival of the fittest, now the survival of the fittest this particular theory was given by Darwin's. So, we all know that he came up with the idea of the survival of the fittest. In short, it says that stronger candidate has a more chance of survival than weaker candidates in an environment of limited sources like food etcetera.

Now, here it is important to know that a word has been written that is more chance this means that stronger candidate has a higher probability of survival as compared to the weaker candidate. And that is why when there is a crossover and a mutation and only the stronger candidates will be survive with the higher probability than this survival of the fittest is going to select only those stronger candidate and will be deleting the weaker candidates.

It is important to know that once when we perform the crossover. So, we are actually performing the crossover on the stronger parents. How? Because the survival of the fittest is aligning only the good solution to be stay and the bad solution which should be deleted.

So, what will happen that if the crossover is apply between the two parents which are survived the survival of the fittest, it means that the offspring that will be created should possess the similar strong traits what the parent is. So, what we can expect that there is a high probability that the new offspring will have a good traits as well as with the mutation these candidates even can be better than their parent solutions.

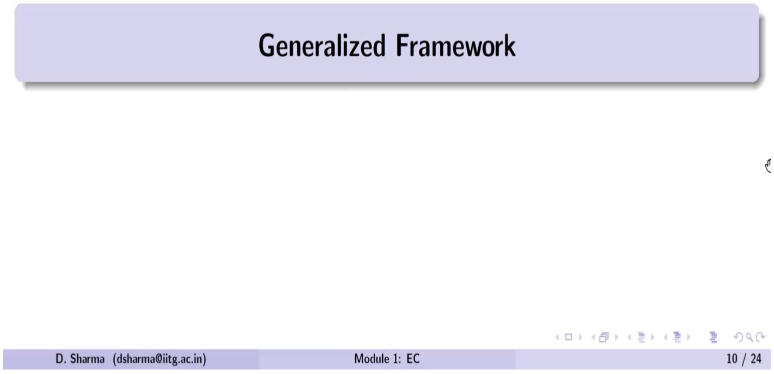
And, that is why we always say that diversity drives changes in the population. So, again the example of evolution of human being where we started moving on the four leg four support and finally, we are now moving and doing our task we need only two supports.

So, that evolution from that age to the current age we have seen like there is a crossover, mutation, the information is coded inside us and after crossover mutation there is a high probability that the stronger candidate were there were retained by the survival of the fittest and that is why generation by generation they were improvement and that is why we are we have the structures currently what we see.

So, generally the question is looking at the evolution so, biological evolution of human being if we say that are we the optimum solutions or sub optimum solution. So, answer to this question is still we are sub optimum solution. The best way to compare is that now if you see yourself and you see a one particular kid of say age 5 or 7.

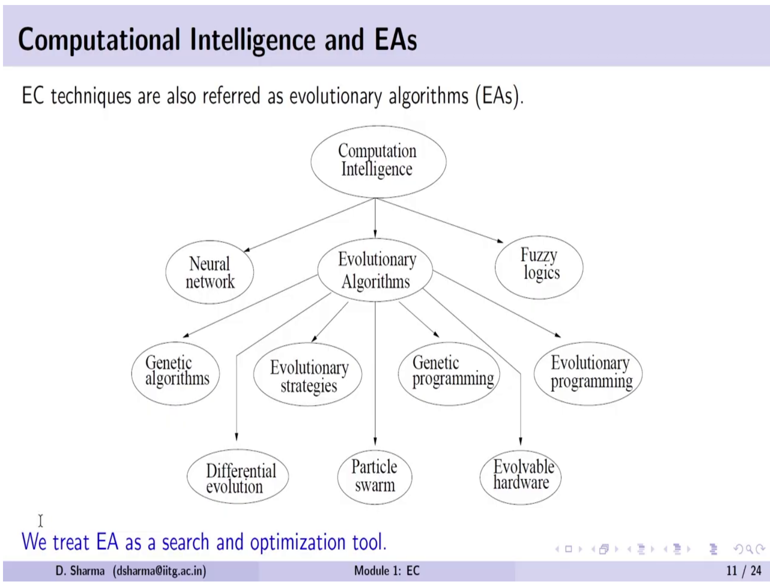
Now, when we were 5 and 7 age, we were not know we were not knowing so many things what these kids know. Similarly, there are other things that have been changed with the time. So, what we can expect is that we are some of the some optimum solutions and generation by generation off springs are improving and that is the best motivation behind evolutionary computation techniques.

(Refer Slide Time: 33:16)



Let us start with the generalized framework. Now, the objective of the generalized framework is we can mention the major steps which are required in the EC techniques and then those steps will be more or less common for other EC techniques.

(Refer Slide Time: 33:34)



So, before we begin that we should know where is if EC techniques lying. So, look at the top I have written that EC techniques are also referred as evolutionary algorithms. So, it EC techniques or EAs they are commonly they are referred to each other. EC technique is a sub field of computational intelligence techniques.

Now, under the computational intelligence technique you can see we have neural network, we have evolutionary computation and we have fuzzy logic as we know neural network we use it for mapping the input verses output. So, kind of a regression.

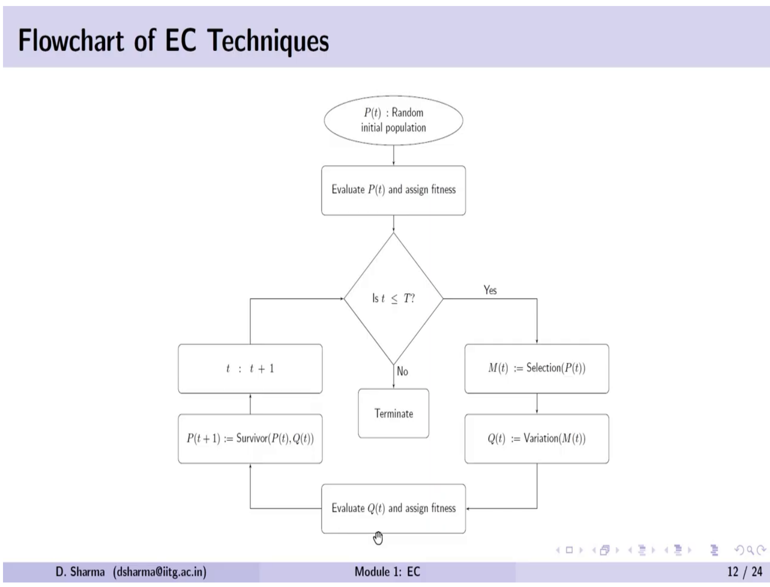
So, based on the data the neural network makes the model and they predict fuzzy logic we use it to represent the uncertainty in our model. However, evolutionary algorithms are used for an optimization and these under the evolutionary algorithm there are many algorithms that have been proposed till now.

Very few of them are genetic algorithm, evolutionary strategies, genetic programming, evolutionary programming – these are the few algorithm which have been developed long back and they were the starting algorithm on which people have started exploring the EC techniques.

Thereafter, we have other algorithm like differential evolution, particle swarm optimization, artificial bee colony, ant colony optimization and cuckoo search and there are so many algorithms if you searches you may get lot of them. However, what you will find they follow the similar principle of EC techniques.

At the bottom, we have written that we treat EA as a search and optimization and that is the our first topic when we started our session 1 that when we are going to optimize something, it involve search and optimization and therefore, we treat EA as a search and optimization tool.

(Refer Slide Time: 35:40)



Now, let us come to the flow chart. So, we will discuss this flowchart one by one. So, we will start with the random initial population. So, it involves three word – random, initial and a population now population as we know that when we in the population we have multiple solution to start with.

So, we generally generate multiple solution in our search space. So, these solutions are generated randomly. So, once we know what are the bound on say x and a y, then we generate solution in that particular space of x and y.

These solutions are randomly generated and since it we are beginning with it. So, it is called initial population. Now, once we generated the solutions with different values of x and a y, we have to evaluate them.

Now, what we mean by evaluation? Evaluation includes finding the objective function value finding the constraints. Now, when we are finding the objective function value and a constraint for different values of x and a y then we have to know which is better or we should give some kind of a fitness.

Now, what we mean by fitness here? See for example, in our class there are lot of students and these students appear in the different exams different subjects. Now, what we do? What we do here is we try to find the aggregate of the marks which a student got in different subjects.

So, finding those aggregate marks will help us to find who is rank 1 in the class, who is rank 7 in the class, who is rank 20 in the class. So, when we say rank one 7 or a 20 we are nothing, but assigning a fitness to a solution. So, in a similar way we have different ways to assign fitness to our solutions.

Once the fitness is assigned we are at the decision box. Now, look at the decision box here the simplest termination condition for our algorithm is based on the t. Now, small t stands for or the numb the counter on the number of a generation and a capital T signifies the maximum allowed generation.

So, suppose small t is smaller than capital T. So, yes, we will go into the box called Selection. Now, here we perform the selection the objective is can we select good solutions. So, that it will create a matting pool or it will create a pool to perform the variation operators.

So, the selection stage here will be selecting few good solutions and storing into M(t). Now, this particular M(t) which we generally say as a matting pool we take it to the next stage because in this matting pool where we have selected few good solutions we perform variation. So, just as an example we have shown that crossover and mutation can be used for the variation.

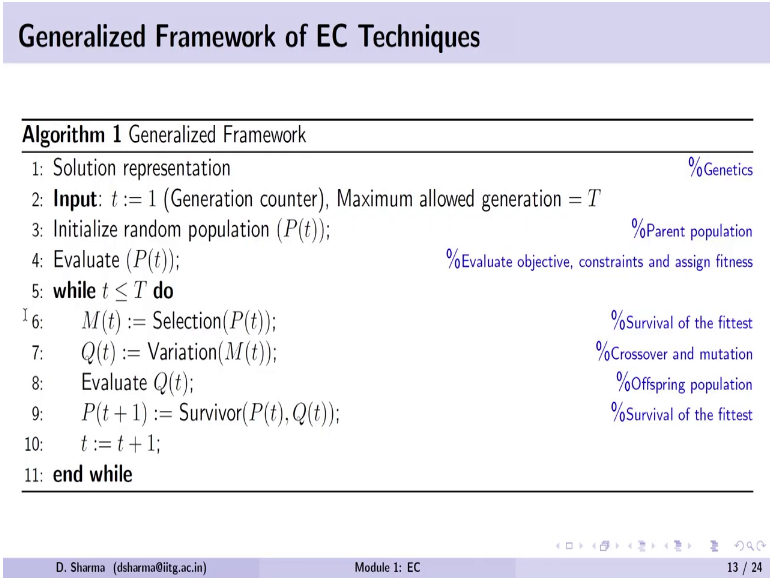
So, when we apply variation on them we will be creating new solutions. Now, once we create new solution we are saving into Q(t). Now, generally this Q(t) is referred as an offspring population and if we go on the top this P generally represent for the parent population. So, we get now coming to the Q(t) this is the offspring population.

What you will realize that once we apply the Variation operator so, the string will change. When the string will change we know that the value of the variable will change accordingly. So, therefore, we have to again evaluate this new population called offspring population and similarly, we have to assign a fitness to each and every solution.

Now, at this at the next stage is called survival stage you might have realized that suppose we starting with say hundred random solutions at the beginning. So, it means that the parent population is made of 100 solutions. At this stage let us assume that after performing the variation operator we have Q which is also made of 100 solutions. So, we have 100 plus 100, 200 solutions at this stage.

Now, there are different ways we can select either the best from both of them or one of them. So, this survival stage will help us to select the best solutions that will move forward to the next iteration. So, in the next box you can see we increase the counter by 1 and we keep on moving in this particular loop we will move till this termination condition get satisfied. Once it is satisfied we terminate our algorithm and we report the optimum solution.

(Refer Slide Time: 40:54)



In an algorithmic way, how this common framework or generalized framework will look like? This representation is important why because this algorithm representation can help us to code or make a program for evolutionary computation techniques.

So, let us start with step 1 which suggest solution representation. Now, solution representation means it can be a binary string as of now, for example. Now, in the step 2, we give certain input to them. As of now I have mentioned the generation counter as a small t and a capital T as a maximum allowed generation. You will realize later that there are certain input parameters that have to be set to start with the EC techniques.

So, all other input techniques which are required to run the EC techniques will be putting in the step 2. In the step 3, we start with the initial random population. As I mentioned we generally create the population randomly and so that we should not be biased towards any of the region or the search.

In the step 4, we evaluate. As you can see here that evaluation means we will be evaluating the objective function constraint and finally, we have to assign the fitness. Now, we are in the while loop and the while loop gets terminated based on the condition given on the number of generation here.

Now, once the if the condition is not satisfied, we are at the step 6. So, the first step in the step 6 we do selection. So, selection is meant for performing say variation operators. So, here what you will realize, that this selection is actually referring to the survival of the fittest.

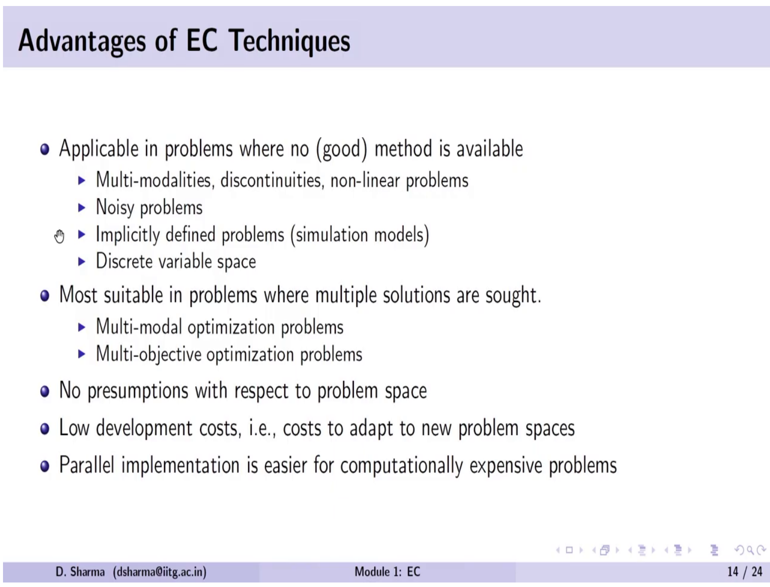
At step 7, we perform variation this means like crossover and mutation and there are other operators associated with other EC techniques. Those operators we apply so that we can change the solution the value of the decision variables. Thereafter we get in the step 8; we get Q(t) which is the offspring population. Since it is a new population we have to again evaluate it and assign the fitness.

In the step 9, we come to the survival stage. So, this survival stage is a motivated from the survival of the fittest that let us choose the good solution that will be moving into the next generation. Thereafter, in thereafter in it is in step 10 we increase the counter by 1 and then we move in this loop.

As you can understand that this loop will be following till the termination condition is satisfied. Once it is termination condition is satisfied we have to stop terminate the algorithm and report the result. So, here looking at the flowchart and the algorithm representation the flowchart is good to understand how our EC techniques work.

The algorithm representation also does the same thing, but it will also tell us if we are going to make a program for our EC techniques.

(Refer Slide Time: 44:19)



Now, we have come to the stage where we want we should talk about the advantages of EC techniques, now it is important to note that if we are working on EC techniques we should know what are the advantages it should offer to us for solving our problem.

So, first advantage is these EC techniques are applicable in problems where no good method is available. So, the problem is so complex or difficult to solve that the methods which are available are not able to give you the satisfactory or a desired solution; in that case it is good that we should start with the EC techniques.

So, what could be the complexities? So, if we remember that we have gone through the properties of the practical optimization problem. So, some of the properties as it is mentioned here that it the function can be multi modal, it can be discontinuous or the function can be non-linear. So, these kind of a of properties or the characteristic of the problem will make the problem so difficult that it may not be solved by other methods.

Problem can be noisy. It can be it can have uncertainty as well. Now, it is important that there are some situations where we have two couple hour simulation model with optimization. Now, when we are coupling the simulation model this means that this is a black box for an optimization algorithm. However, for an optimization algorithm there is no equation on which it will work.

So, what EC techniques will does? EC techniques will do that they will pass a certain input to the black box. Black box will do some kind of a simulation for you and give you the output and these output we can use it for finding what should be the objective function and constraint and finally, assigning the fitness.

Now, here it is important to realize that such kind of a black box will not allow to have any mathematical equation. So, therefore, so the problems which are have which do not have any strict mathematical equations, in that situation EC techniques can also be used. Come to the discrete variable space, these problems are difficult to solve, there EC techniques show an upper hand over the other algorithms.

There is a situation when we will be looking for multiple optimum solution. So, what could be those scenarios? Multi modal. So, remembering the plot for a multi modal problem where for a different values of x we have maximum of the function.

Since we are interested to find all these values so, the EC techniques will help us to identify those different kind of optimum solution. It is only because EC since EC techniques work with the population. So, different solution can be concentrated on different peaks, so that we can identify almost all the peaks which are related to multi modal problems.

Multi-objective optimization as you know that when we have more than one objective. So, in that case there is no single solution which is optimal. There is a set of solutions which are optimal and which we generally refer them as Pareto optimum solution. Since evolutionary computation techniques work on the set of points so, that is why it is called population based algorithm. So, these solution will help us to find those solutions which are optimal or Pareto optimal solutions.

One good important point with the EC technique is that there is no presumption with respect to the problem space. So, the problem space is continuous or a discrete, it is non-linear or the problem space has different kind of variables, there is no presumption. You need to change the type of variables objective function and a constraint to the problem and the EC techniques will be able to solve such kind of problem as well.

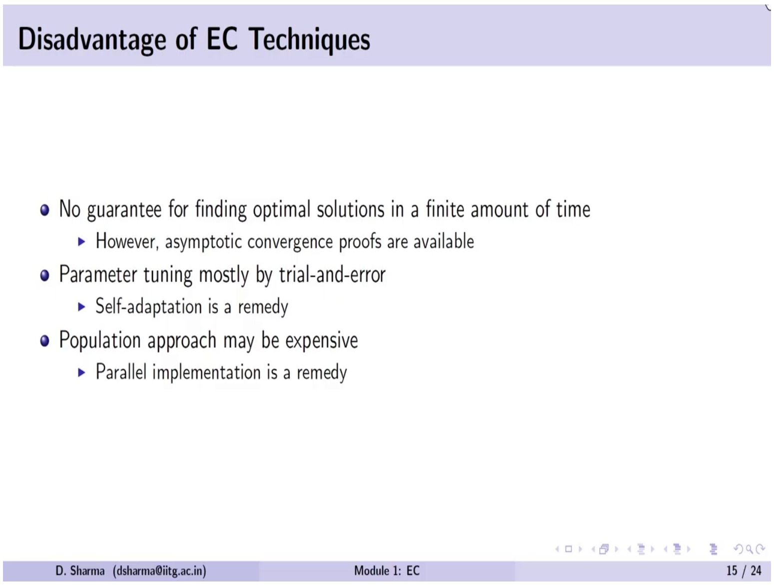
Another very good important property or advantage is that there is a low development cost to adopt to a new problem. Say for example, the one of the EC technique is made and it is working fine for a one particular application in say electrical engineering.

Now, what is happening is that since it is working good so, people working in a different domain in other engineering and science for example, mechanical, computer science, civil, electric, chemical engineering, sometimes physics and a chemistry.

So, you have to take the same code, just change the objective function constraint and the type of variable you can use it. So, rest of the operators will remain the same. So, that is why we say we have a low development cost if I want to use the same algorithm for different problems.

Come to the parallel implementation. Now, this parallel implementation is important. It is because when if the problem is computationally expensive it is taking too much of time to get an optimum solution. So, why do not we paralyze it? So, if we make it parallel this means that the function evaluation will be done in parallel that is finally, going to reduce a time for us.

(Refer Slide Time: 50:03)



Now, with these advantages that gave us a lot of motivation to use EC techniques for our problems. However, probwe should also know that what could be the disadvantages of EC techniques. So, there are certain limitations which we should know before we apply.

The first disadvantage is, there is no guarantee for finding an optimum solution in a finite time this means that we do not have any proof available that will say that EC techniques will converge to the optimum solution in a finite amount of generation or iteration. Since, there is no proof available many researchers, they do not want to use it.

However, since there are certain advantages which we have discussed in the previous slide that allows and motivate us to use EC technique for solving complex problems. It is important to know that even though there is no guarantee, but there are some asymptotic proofs available that says that if I take a larger population size, if we run the algorithm for a longer time we may reach to the optimum solutions or close to the optimum solutions parameter tuning.

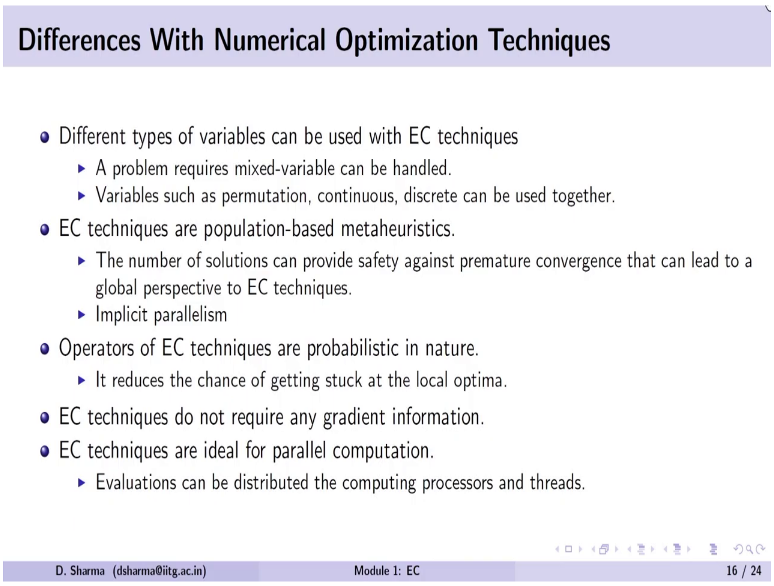
Now, it is a disadvantage why because every evolutionary technique has some user defined parameters. Now, those decision those user defined parameters we have to set before we run the algorithm. So, you may find that a one particular value of a user defined parameter those parameters will work for certain class of a problem, but for other class of a problem you have to change those input parameters.

So, that is why setting those input parameters or tuning them is a difficult task and generally we do it trial and error. So, that possess a challenge for us how to set them. The remedy to remedy to this particular problem is the self-adaptation. So, what we can do here is we can learn from the value of say objective function we can tune those parameters by its own, so that our algorithm should converge to the optimum solution.

The third disadvantage since it is a population based meaning that we have multiple solutions to start with and for every solution we have to find what is the; what is the objective function as well as constraint. So, in that case we are actually computing too much. So, that makes our problem computationally expensive.

So, with these disadvantages there are certain remedies which we can use it. One of the remedy is we can use parallel implementation, so that we can reduce the overall time here.

(Refer Slide Time: 53:00)



Coming to the differences with the numerical optimization technique. Now, as you know that numerical optimization techniques are developed based on the mathematical rules. So, the first difference which you will find is that if in a problem there are different kinds of variables.

It can be real number, it can be a binary, it can be a discrete number, if all those mixed variable problem is there EC techniques can easily handle them. However, if you look at the numerical optimization techniques these are not easy to implement because every type of variables requires different ways to handle it, but EC techniques can easily handle them and can solve such kind of multi variable problems.

Second is EC techniques are population based. So, we have seen the example that we always have set of solutions from which we perform say variation operator survival of the fittest, so that we eventually get the optimum solution. But, in most of the numerical optimization techniques they have either one solution or very limited solutions. So, that is the another difference which we can find it.

Now, what is the advantage we can get it from the population based metaheuristics? Now, the number of solution basically provide safety against premature convergence that can lead to a global perspective to EC techniques. Now, you will realize that when the number of a solutions are there and when we are performing crossover and a mutation or any other variation operator. So, they we are exchanging some information. When there are multiple solutions which are lying in a different region of the search space there are less chances that our algorithm will prematurely converge to the local optimum.

However, with the numerical optimization technique it can be a problem because generally the direction where we move it is decided based on the local perspective. Since we have multiple solutions to work with them so, most of the time EC techniques are considered as a global optimizer.

Second property is that we have a implicit parallelism now this parallelism is means that the set of points they work together perform crossover and a mutation. So, you will realize that these solutions they are mutually working and moving towards the optimum solutions. So, that they have a parallel implicit parallelism to move towards the optimum solutions.

The third difference which we can identify is the probabilistic nature of the operators which we use in EC techniques. So, as and when we will discuss the algorithm you will realize that the operators are stochastic in nature. Since they are stochastic in nature so, there is no fixed rule.

We make rules, but those rules are run based on the stochastic behavior of the operator. However, in the numerical optimization we have mathematical rules and those rules are applied iteration by iteration to get an optimum solution. However, in the EC techniques these are probabilistic or stochastic in nature.

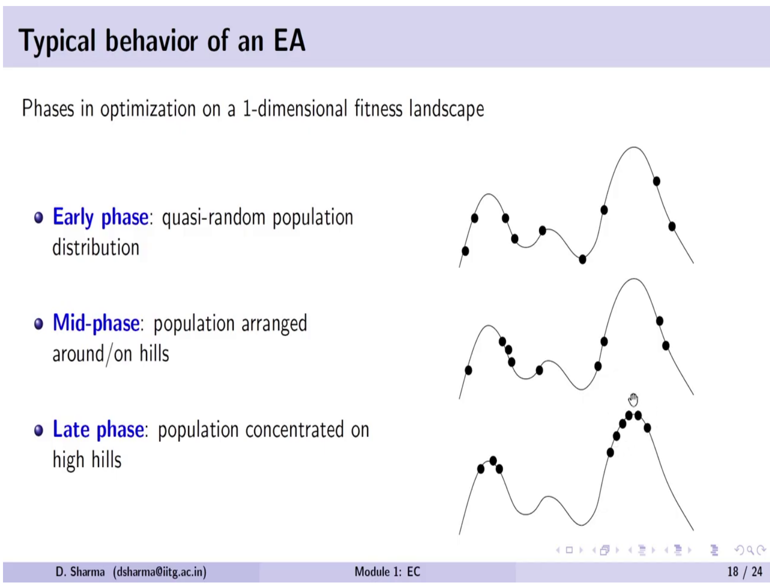
What is the advantage of stochastic nature or probabilistic nature that it reduces the chance of getting stuck at the local optima. So, it is only because since it is probabilistic in nature, we may generate solutions away from the local optima or may be at the distance places.

EC techniques do not require any gradient information. This is applicable to all those techniques which require gradient. Now, since we evaluate the objective function and constraint and we assign a fitness we do not need any gradient information. So, that reduces the complexity for solving a problem which are discontinuous and there is or the function is not differentiable.

So, in that case EC techniques since it is EC techniques do not need any gradient information, we also sometimes call them as EC techniques for a as a direct search methods. Then we have EC techniques the another difference is the EC techniques are ideal for parallel computation.

As we have discussed earlier that since we are working on multiple solutions these solutions will be these solution will can be computed in parallel on different processors. So, that will help us to reduce the overall time. However, with the numerical optimization technique since we work only with the single solution. So, we do not have any scope to paralyze them and reduce the computation time.

(Refer Slide Time: 58:17)



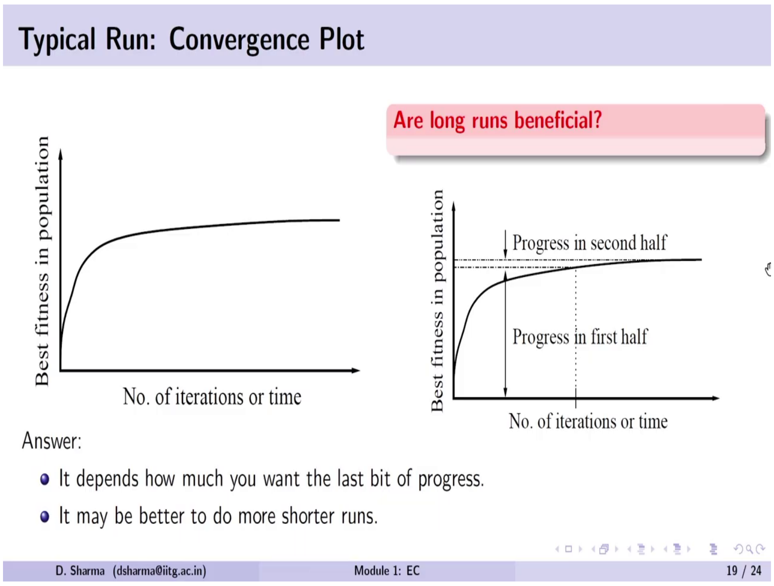
Now, let us move towards the typical behavior of EC techniques, now for understanding the typical behavior we have taken a 1-dimensional fitness landscape here, a simple problem to solve, but looking at the landscape here you can see that you have global optima and you have local optima.

So, let us suppose that we want to maximize the function. So, in the early phase as you can see here in the early phase we generally generate the points randomly. So, we call it initial population.

So, the distribution of a solution on a landscape on the right hand side figure you can see that the solution are distributed on the landscape. After performing set of crossover mutation and a survival of the fittest you will realize in the mid phase these solutions will be moving slowly towards the peak.

So, these set of solutions are moving to this peak, this solution will be moving to that peak, these four solution will be moving to the third peak. Finally, at the last stage what you will realize that the solution will be adjusted to the different peaks and we can get the optimum solution.

(Refer Slide Time: 59:25)



Now, let us talk about the typical run. Now, in the typical run you can see the convergence plot what you will realize that the best fitness is written in the y-axis and the number of iteration or a time in the x-axis. So, there is a lot of improvement at the initial stages and afterwards the improvement is not much.

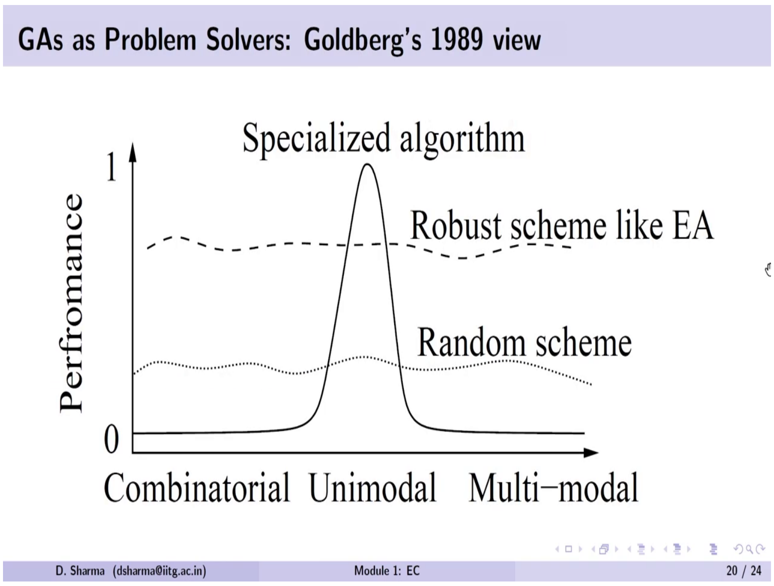
Now, question is long runs beneficial? So, what we have done in this particular situation we allow our algorithm to run for a longer time. Now, look at the figure on the right hand side you will realize that if I divide this x-axis into the two part; so, this is called first part and this is called and second half.

Now, in the first half this is the improvements, best fitness for a maximization function we can see there is a lot of improvement. Thereafter in the second phase the improvement is very little, you can realize from a small portion here. Now, question is do we need to run for a longer time or we are happy as soon as we get the better solution than the existing.

So, here we have two approaches: one approach is called engineering approach – in this case we are happy as long as the solution is is better than the existing one. So, we can stop our algorithm. Another is called mathematical approach in which the we are looking for exact optimum solution.

So, in that case me we may want to run our algorithm for a longer time, so that we should get an optimum solution. So, the as the answer to the question is it depends how much you want the last bit of progress. So, how much precision you want in your solution we decide our iteration or a time to decide the progress. Similarly, it may be better to do shorter runs that is suggested.

(Refer Slide Time: 61:21)



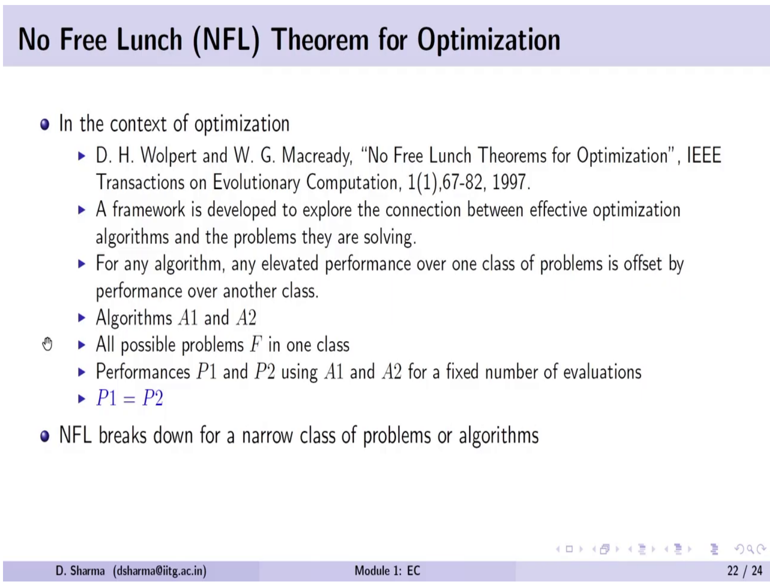
This is a Goldberg's view for GA and in this particular figure we have generalized it for the evolutionary algorithm. Now, you can see on the x-axis there are different kind of problems like we have combinatorial problem, unimodal and a multi-modal.

Now, you will realize that there are certain specialized algorithms. So, those specialized algorithms will be working 100 percent for their class. See for example, for a unimodal problem these specialized algorithms are the best. However, their performance for other kind of a problem is very poor.

If we take any random scheme to solve such kind of a problem, you can see the performance is more or less same of this random of any random scheme for solving the variety of a problem. Look at the evolutionary algorithm. If we are going to solve them, you get such kind of a performance which is kind of equivalent to the random scheme.

What is the difference between these two? Random scheme is below average and our evolutionary algorithm is quite above average and that motivate us to use evolutionary algorithm for those problem whereas where we do not know where is the optima or difficult or the problems which are difficult to solve.

(Refer Slide Time: 63:04)



Now, let us come to the no free lunch theorem. This theorem is important in the present context because when we are going to learn different algorithm so, we will get to know or we should understand that these algorithms will be useful for what class of problems.

So, in this no free lunch theorem so, these two researchers they come up with this theorem for optimization. A framework was developed to explore the connection between the effective optimization algorithm and the problem they are solving. As I as we said that since we will be learning EC techniques so, we will get to know that what kind of problems we can solve them.

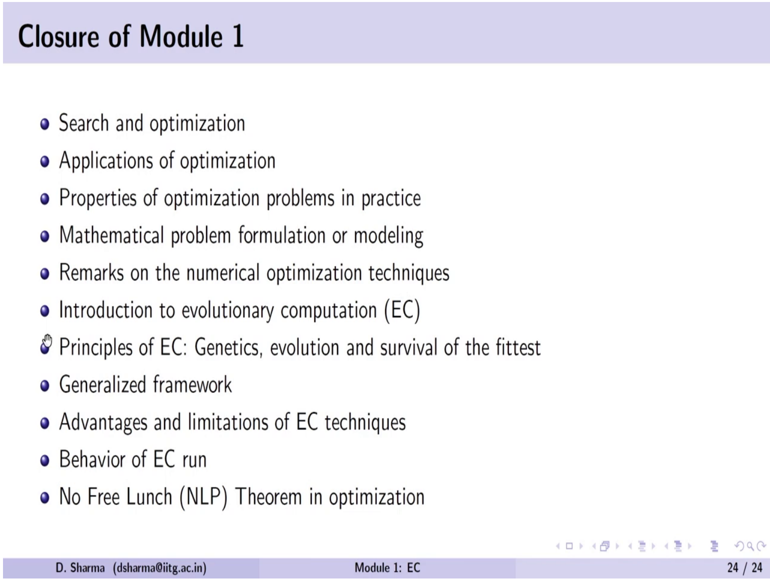
So, for any algorithm, any elevated performance over one class of a problem is offset by the performance over the other class. So, what is happening is, suppose we are using we develop one EC techniques and we are solving a one set of a problem and in that when we are looking at the performance it is working fine, but when we are solving another set of a problem the performance may not be good.

So, in this case the performance in one set will help us to decide this algorithm will be useful for certain class. So, let us take an example here suppose we have algorithm A1 and A2 and these two algorithms are run on a one class of a problem which is called F. Now, while solving a one class of a problem of using the fixed number of evaluations we get performance P1 and a P2, we can say that the algorithm 1 is equivalent to algorithm 2. Solving a one class of a problem for a fixed number of evaluation when their performances are the same.

So, therefore, when we compare we have to be we have to do a fair comparison while choosing the set of problems as well as the function evaluation. So, the overall advantage of no free lunch theorem is that it breaks down for a narrow class of problems and algorithm and that is important.

Why because, once we know the characteristic of the problem we can decide which algorithm is going to help us to solve that problem. So, therefore, the research effort is to find the best algorithm for a class of a problem. So, the problem can be unimodal, multi modal, quadratic etcetera.

(Refer Slide Time: 65:30)



With this I will I want I would like to conclude this session. So, in this module 1, we have gone through the search and optimization where we discussed about how we start with a set of variables we keep on searching and finally, we get the optimum solution. So, that is why we call together as opt as a search and optimization.

Thereafter we talk about the applications of an optimization ranging from mechanical, civil, electrical and various other fields. We talk about those applications to motivate us or for the for looking for the optimum solution on those complex and real world problems.

We also talk about the properties of the problem because these properties will help us to decide what complexity it has. So, it can be discontinuous, it can have non differentiable function, multi modal, robust or reliable solutions. Then we talk about the mathematical formulation, a generic representation of mathematical equation for objective function, various types of constraints, variable bounds that were discussed.

During that time, we also discussed how to choose decision variables, how to make constraints, type of constraints, the objective functions and variable bounds. Finally, we talk about the remarks on numerical optimization. Those remarks motivate us to come up or to look for an algorithm which is flexible and can solve large class of problem.

In this particular session, we have gone through the introduction of evolutionary computation, we talk about EC techniques, we talk about the principle of EC techniques such as genetics, evolution and the survival of the fittest. So, all these principles we discussed with an example, so that we can understand that how these EC techniques are going to work.

We discussed about the generalized framework in the form of a framework and the algorithmic way. The framework was help helped us to understand the working principle of the EC techniques and the algorithmic representation will help us to write a code or representing the algorithm in an algorithmic way.

We talk about the various advantages and limitations. So, when we are solving we should know what kind of where we will get the advantages because there are certain class of problem for example, quadratic problem. Since quadratic problems are easy to solve by numerical optimization we do not need EC techniques to solve them. Therefore, it is important to know what certain class of problems, complex problem, EC techniques will be useful.

Similarly, we talk about the limitations. For example, there is no proof for the convergence in the limited number of time or iterations or there are lot of user defined parameters with EC techniques. Thereafter, we discussed about the behavior of EC techniques.

So, behavior on a one-dimensional landscape help us to understand that the solutions were distributed throughout the range of x and with crossover and mutation and survival of the fittest applied for certain number of generation, the solutions were reaching or moving towards the peak.

And finally, in the final stage the solutions are converged to the peak of the solutions and finally, the no free lunch theorem it is important because when we are learning, when we are making our new algorithm after going through this course you will realize that when we make an algorithm we have to make a fair comparison.

For fair comparison we should know what class of a problem this algorithm will be useful and what are the other algorithms I we can solve it. All these algorithms should be compared on a common platform. For example, we can fix the number of function evaluation and let us see how these algorithms perform on a set of problem.

And, then we can compare their performance and then we can suggest whether this algorithm is better than the other algorithm or a not. With these notes, I would like to conclude this module 1.

Thank you very much.