

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

Lecture - 13
Faster Genetic Algorithms

I am going to start with another topic, that is topic 7 and it is on Faster Genetic Algorithms. Now before I go for this, let us try to understand that why do you need faster genetic algorithm or why do you need faster optimization algorithm. Now, supposing that I am just going to automate one engine process or one engineering system. Now if I want to automate that; so I will have to take the feedback, and this particular feedback will have to take within a fraction of second and supposing that I am using one optimization tool just to develop that particular; that automated process.

Now if I use one optimization tool and if it takes a considerable amount of time to give that optimal solution, so I may not be able to automate that particular the process. So, if I want to use one optimization tool; so in this particular automation process I will have to ensure that this particular optimization tool can yield the optimal solution or the near optimal solution within a fraction of second. And that is why the concept of faster genetic algorithm or the concept of faster optimization tool came.

Now if you see genetic algorithm it is computationally very expensive and it takes a long time; a large number of iteration to give that optimal solution.

(Refer Slide Time: 02:00)

Micro-GA

- ❖ Faster GA having a small population size
- ❖ Proposed by Krishnakumar

Steps

- ❖ **Step 1:** Select a random initial population of binary strings of size 5
- ❖ **Step 2:** Evaluate the fitness of the strings. Mark the best string as string 5. Copy it directly into the mating pool → Elitist strategy

Handwritten annotations:
- A box labeled "Mating pool" is drawn around the text "Copy it directly into the mating pool".
- A box labeled "Imit." is drawn around the text "Elitist strategy".
- A diagram shows a population of 5 binary strings: 10, 01, 11, 01, 11. Fitness values f1 through f5 are written next to them. An arrow points from the string "11" (string 5) to the "Mating pool" box.

Now if I use the conventional genetic algorithm in the process of automating one engineering system, the very purpose of using that optimization tool may not be solved because it will take a long time. And that is why people thought that can you design and develop faster genetic algorithm.

Now, micro-GA is actually a result of, so that type of thinking. And this is a faster GA. And here we start with a small population size and generally we consider the population size to be equal to an odd number. So, typically we consider the population size that is capital N is equal to 5 7 and so on. And this concept of micro-GA that was proposed by Krishna Kumar, and after that it became very popular particularly for the problem where we want to find out the optimal solution within a fraction of seconds.

Now this is basically a binary coded GA with little bit of modification. Now those modifications I am going to discuss in detail. Step 1: we select a random initial population of binary strings of size 5. So, as I told that we start with a population size n equals to 5 and that is an odd number. Now, as I told that either 5 or 7 and 9 generally we do not go more than 9 in this particular the mike regime.

Then step 2: we evaluate the fitness of all 5 GA string line in the population and initially those are selected at random. Now in binary coded GA actually or any other GA what we do is we try to form the mating pool starting from the initial population. And in the mating pool we try to copy all good solutions. Now what we do is out of 5, so we try to

find out which one is having the best fitness. Supposing that, I am solving one maximization problem. So we try to find out, so out of these five which one is having the best solution best fitness; so that particular best string is marked as string 5 and copied directly into the mating pool.

Now, if I just want to explain further so what we do is; we create one initial population of size 5 say one could be 1 0 1 1, 0 1 1 0, 1 1 0 0, 0 1 1 1, 1 1 0 1. Now here supposing that I am solving one maximization problem; now we can find out the fitness values very easily f_1, f_2, f_3, f_4, f_5 . Now out of these five fitness values which one is the maximum for a maximization problem you mark, supposing that, so this particular the string is going to give the maximum fitness value; so this fitness value, this particular GA string that will be directly copied to the mating pool. Say if this is the mating pool, say this is the initial population and this is the mating pool; so this particular the third string that will be directly copied as string 5. And I am just going to copy here. So this 1 1 0 0, because this is having the highest fitness and that is directly copied in the mating pool. This principle is known as the Elitist Principle.

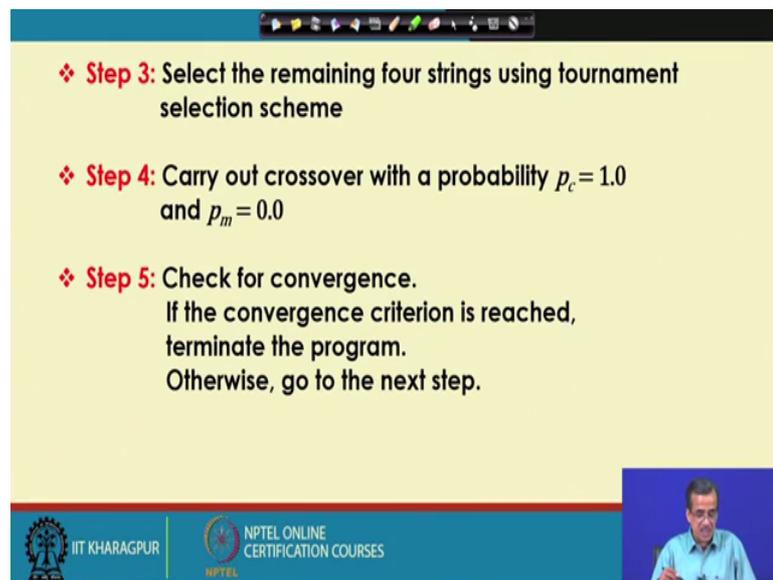
That means, the already found base solution we give a direct copy in the mating pool. And once you have got this particular fifth string, now I have got fourth vacant position in the population. Now to fill up these four vacant positions what we do is, we take the help of tournament selection. Supposing that, we select the tournament size of 2; that means, at a time out of this particular five any two solution will be selected at random and we try to find out the better in terms of fitness. And here we will have to remember the already found based solution which has been copied directly here as elite will also participate in this particular tournament selection.

So, if I play tournament for once I will be getting one GA string here, similarly I will have to play four times this particular tournament so I will be getting 4 GA string here. So, this is nothing but my mating pool. And once I have got this particular mating pool, now I will just go for the next operator that is your crossover. Now before I go for this particular crossover what we do is- my fifth string is not going to participate in crossover it is just like the fixed deposit. So, directly it will be copied to the next population. And the remaining four they are going to participate in crossover. So, using this 4 GA string, so I will be getting only two mating pairs. And supposing that the probability of

crossover is equals to 1.0. That means both the mating pairs are going to participate in this particular the crossover.

Now, so here there will be crossover and these two mating pairs which will be selected at random out of these 4 GA string they are going to create the children solution. And this particular fifth string is already there, the elite is already there, so what will happen is I will be getting; the next thing is actually.

(Refer Slide Time: 09:03)



❖ **Step 3:** Select the remaining four strings using tournament selection scheme

❖ **Step 4:** Carry out crossover with a probability $p_c = 1.0$ and $p_m = 0.0$

❖ **Step 5:** Check for convergence.
If the convergence criterion is reached, terminate the program.
Otherwise, go to the next step.

So, I will be getting the population after the crossover and we generally do not consider any mutation in this particular the micro-GA, because this is a small population GA. Now step 5: we check for convergence. If the convergence criteria is reached, we terminate the program. Otherwise, we go to the next step.

The next step is as follows.

(Refer Slide Time: 09:32)

❖ **Step 6:** Create a new population of strings of size 5 by copying the best (elite) string of the semi-converged population and then generating the remaining four strings at random. Go to Step 2.

The slide features a diagram with two boxes. The left box, representing the 'semi-converged population', contains five binary strings: 01-00, 01-01, 01-01, 01-01, and 11-01. A red arrow points from the first string (01-00) to the right box. The right box, labeled '2nd gen', contains five horizontal lines representing a new population of strings. The slide also includes logos for IIT KHARAGPUR and NPTEL ONLINE CERTIFICATION COURSES at the bottom.

We create a population of new string of size 5 by copying the base string of the semi-converged population and generating the remaining four at random.

Now supposing that, I am getting after this particular the crossover the 5 GA string as followed. So let me draw the elite first, say this is the elite and other things let me write at random 0 1 1 0, 1 1 0 0, 0 1 0 1, 1 0 1 1. Supposing that this particular population I am getting after the crossover and there is no mutation, so this completes one generation or one iteration of this particular the GA.

Now, if the termination criteria is not reached we go for the next iteration. Now to go for the next iteration what we do: we compare the fitness of this particular the GA string lying here and supposing that out of these five say this particular thing is going to have the maximum fitness, even better than this elite solution. It may happen that due to crossover will be getting one child solution which could be even better than elite. Now supposing that this is found to be the best, so we directly copy this particular best solution in the population; and let me copy it here 1 1 0 0 and here I am just going to start the second iteration or the second generation.

So at the beginning of second iteration, this particular solution will be directly copied because it is having the best fitness. And the remaining four will be generated at random using the random number generator. Now we use this random number generator to

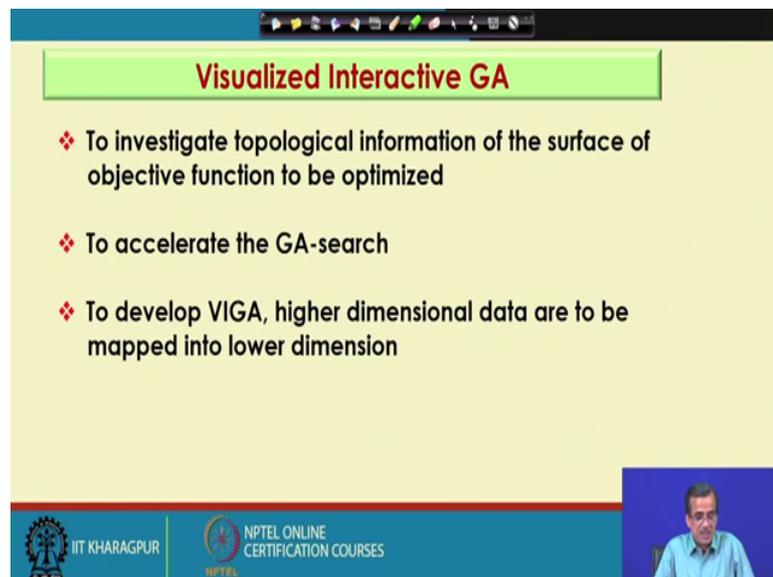
generate this particular the 4 GA string the reason is this will give some sort of diversification to the solution.

Now if I do not select these 4 GA string at random at the beginning of next generation there is a possibility, there will be a chance of premature convergence. Because, this is a small population GA. Just to avoid the premature convergence at the beginning of second generation, so these four solutions will have to generate at random. Then once again we go for the mating pool. For mating pool we use the principle of the elite solution; that is the elitism principle. And once again, once you have got this mating pool using the principle of elitism and we use the tournament selection we go for crossover, and generally we do not use any such mutation here.

Now, this process will go on and go on through a large number of iteration. In fact, we do not need a very large number of iteration might be within 15-20 iteration or generation we will be getting some optimal or the near optimal solution with the help of this type of micro-GA.

So, this micro-GA can be used actually online to find out the optimal solution.

(Refer Slide Time: 13:23)



The slide is titled "Visualized Interactive GA" in a green box at the top. Below the title, there are three bullet points, each preceded by a red diamond symbol:

- ❖ To investigate topological information of the surface of objective function to be optimized
- ❖ To accelerate the GA-search
- ❖ To develop VIGA, higher dimensional data are to be mapped into lower dimension

At the bottom of the slide, there are two logos: the IIT Kharagpur logo on the left and the NPTEL Online Certification Courses logo on the right. A small video inset of a speaker is visible in the bottom right corner of the slide area.

And consequently actually, this concept gained much popularity. Now here actually I am just going to discuss another faster genetic algorithm that is called the Visualized Interactive Genetic Algorithm. Now, to discuss the concept of the visualized genetic

algorithm the main purpose is to make the GA search faster. Now, how to make it faster; that I am going to discuss.

In fact, the proposal came for us only for this particular the visualized interactive GA. The purpose was to investigate the topological information of the surface of objective function to be optimized. Now, supposing that I have got an optimization problem of only one variable. So, y is a function of x , so the objective function is in 2 dimension we can visualize. Let us consider another optimization problem: y is a function of x_1 and x_2 ; sorry y is a function of x_1 and x_2 . So, this is a function of two variables, so the surface of the objective function is in 3 dimensions. But, if I consider that y is a function 10 variables the objective function will be in 11 dimensions; we will not be able to visualize. Because, we human being we can visualize only up to 3 dimension more than 3 dimension like 4 dimension 5 dimension and so on we cannot visualize.

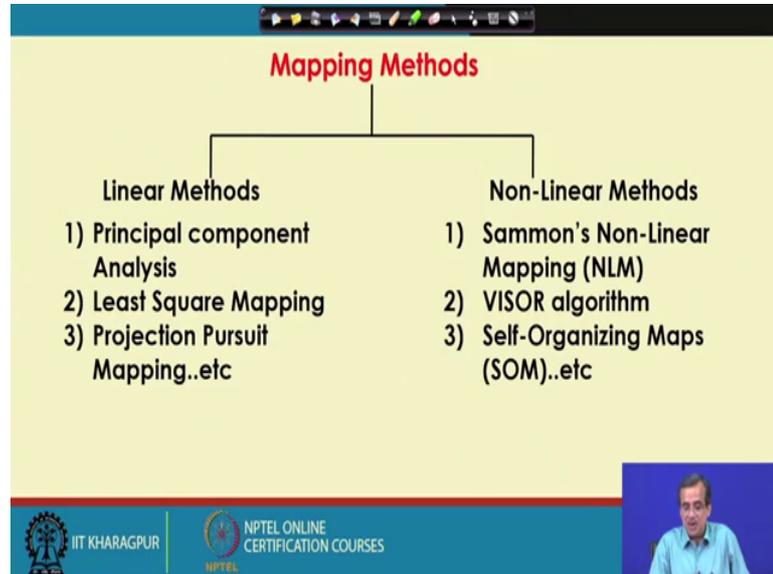
So, if we want to understand the way the optimization tool is moving towards the optimal solution through the various iterations and if you want to visualize that we will fail if it is a function of like 3 or more than 3 variables. Now, for this particular optimization problem where objective function is a function of 3 or more than 3 variables it will be bit difficult to visualize the surface of the objective function. So, here in visualized interactive GA actually what we do is; we try to visualize the surface of the objective function. And if it is in a higher dimension like 3 or more than 3 then what we will have to do is we will have to map it to the lower dimension for the purpose of visualization.

Now, let me take a very simple example. Supposing that, y is a function 10 variables the objective function is in 11 dimension which we cannot visualize. So, from 11 dimension the data; data means the data which are lying on the surface of the objective function I will have to map it to either 2 dimension or 3 dimension for the purpose of visualization. That means, to develop this visualized interactive GA in short which is known as VIGA. So, we will have to use the concept or the methods of mapping from higher dimension to the lower dimension.

Now, the main purpose is to accelerate the GA search so that the GA can reach that optimal solution quickly. Now, actually as I mentioned to develop this visualized

interactive GA, so we will have to use a few methods which may help us to map the high dimensional data to lower dimension for the purpose of visualization.

(Refer Slide Time: 17:27)



Now if you see the mapping methods, we have got a large number of methods available. For example, we have got a few linear methods like principal component analysis which is very popular, then least square mapping, projection pursuit mapping. So, these are all linear mapping methods. And here we will be getting one is to one mapping. On the other hand we have got a few non-linear methods for mapping from higher dimension to lower dimension. And if you see the literature once again a few methods are available. For example: Sammon's non-linear mapping is very popular out of all these non-linear mapping methods, then we have got the graph based technique like visor algorithm, we have got a very efficient neural network that is called self organizing map and others.

Now here, before I proceed further and before I am going to discuss the working principle of this visualized interactive GA, I am just going to concentrate the working principle of some of the non-linear methods; non-linear mapping methods and I am just going to discuss the principle of all these non-linear methods in detail.

Thank you.