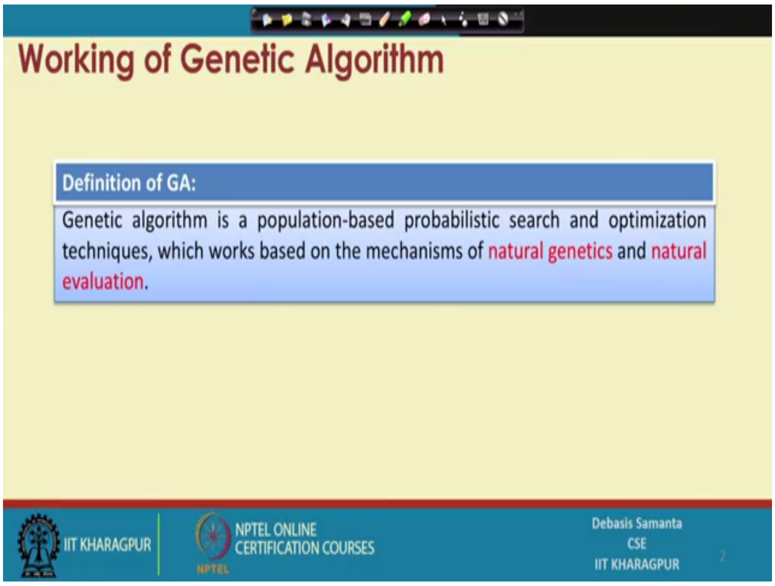
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 15
Concept of Genetic Algorithm (Contd.) and GA Strategies

So, so we are discussing about genetic algorithm, and we have discussed about that 2-biological process namely genetics and evolution, how they can be used to solve optimization problem in the form of evolutionary algorithm called the genetic algorithm. And based on this concept we can define the GA as a population based random search, this means from one population to another population we have to search for the base solution.

(Refer Slide Time: 00:53)



Working of Genetic Algorithm

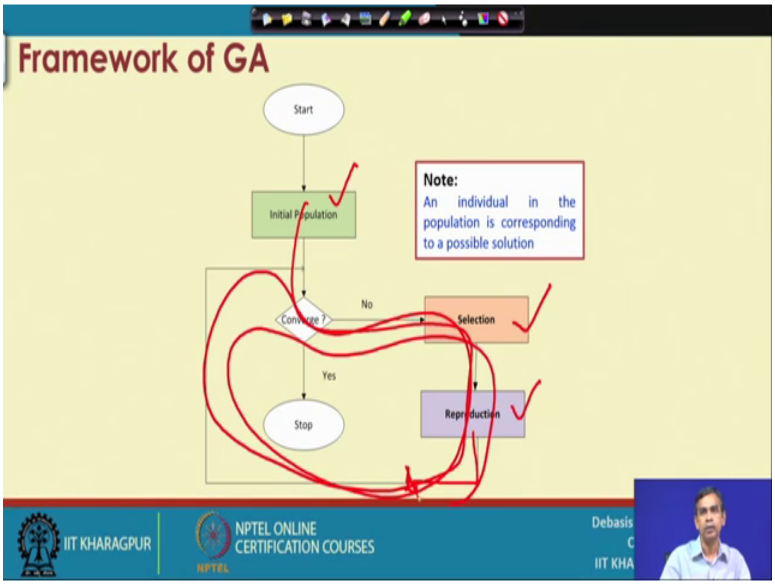
Definition of GA:

Genetic algorithm is a population-based probabilistic search and optimization techniques, which works based on the mechanisms of natural genetics and natural evaluation.

So, it is called the population based probabilistic search, and that is used to solve optimization problem, and such a population-based probability search is based on the concept of natural genetics, and then natural evolution.

So, today we will discuss about details about it. So, what does a population means, and then the random search how it can be carried out by means of the genetics, and then natural selection or evolution.

(Refer Slide Time: 01:27)



Framework of GA

Start

Initial Population

Converge ?

No

Selection

Reproduction

Yes

Stop

Note:
An individual in the population is corresponding to a possible solution

So, the algorithm follows a framework which is shown here, and in this framework, we see 3 basic building blocks the first is the called the population, and then is the selection, and then reproduction, and then these are the basically control to process I mean towards this one. So, this basically eventually is a searching process and is a probabilistic searching process actually, now here few things are important the population, the selection, and then reproduction. So, the selection is basically follows the concept of evolution proposed by Charles Darwin, and reproduction is the concept it is basically the concept of genetics proposed by Gregory Johan mandala.

So, they basically the idea that is their idea is basically how from a current population initially it is initial population the process can be moved. So, that it can gives a new population here, and then it can iterate the same thing until we can have the search complete this is called the converge and finally, we get the solution it is called the optimum solution. So, essentially it is basically starting to this one we have to obtain from here, but by means of this process this process so, long the conversation is complete and we can get it.

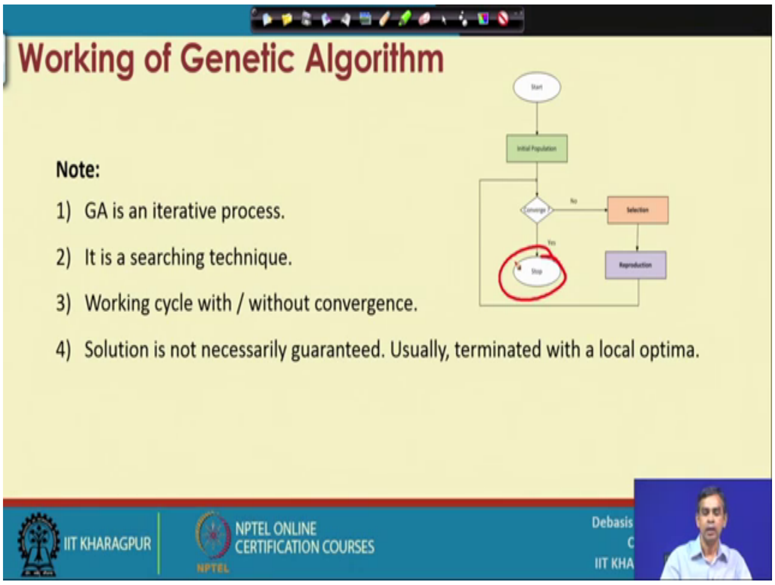
So, the idea it is like this, and this kind of concept it is basically the genetic algorithm concept, and this is the basic framework of the g a. And here one thing that population here is basically is a individual, and this population can be linked to is a solution. Or in other words a population or an individual belongs to a population basically is a possible solution now this is very important and individual in this population is basically a possible solution, these 3 things are very important.

So, somehow if we can represent a possible solution, by means of an individual, and a number of individual is selected and then it gives a generation that is called a population, and from this generation if we follow this selection and reproduction operation so, that we can get another population or another generation then, and this another generation is significant in terms of the improved solution.

So, if it gives a solution and it gives another population, who is consists of improved solution, then if we iterate this one and then by this iteration iterative process, we can get from one solution to more improved solution, and we can stop this searching if we satisfy some convergence criteria; that means, when we have to stop this seemingly is basically the infinite loop.

So, this process is basically followed in genetic algorithm and to understand the genetic algorithm we have to understand, how this initial population can be generated; that means, a possible solution can be converted to an individual, and then how the selection, and then how reproduction operation can be carried out, or can be realized we will learn all these things in the subsequent lectures one by one, first we will discussed about this framework the framework of GA and it is variation. So, in this lecture we will limit our discussion to that only it will help us to understand the concept further, and then later on we will discuss the other concept.

(Refer Slide Time: 05:46)



Working of Genetic Algorithm

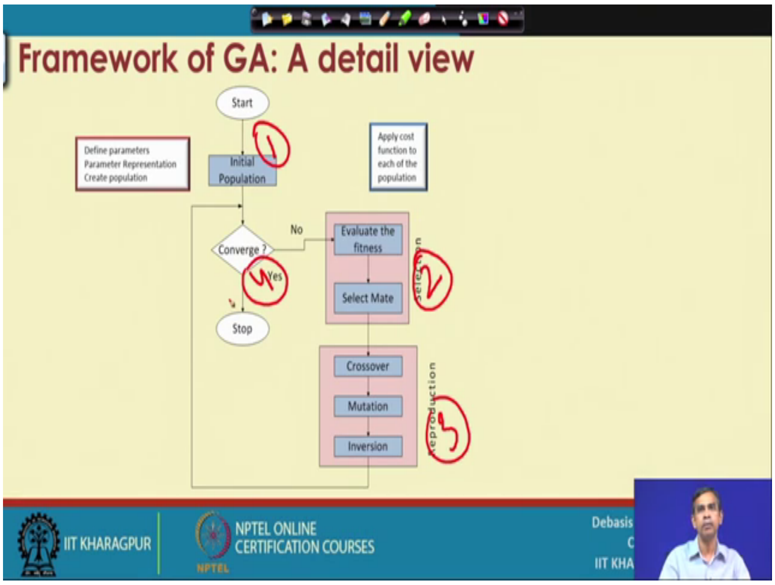
Note:

1) GA is an iterative process.
2) It is a searching technique.
3) Working cycle with / without convergence.
4) Solution is not necessarily guaranteed. Usually, terminated with a local optima.

So, let us proceed here. So, as we have learnt it the genetic algorithm works as an iterative process, and in this iteration it basically pursue the searching, searching for the base solution, and this working cycle is basically continued with certain convergence criteria; that means, whether or you have to continue the search, or you have to stop it, now one important thing is that solution that it ultimately gives here, the solution not necessarily to be guaranteed, but it can give near optimum solution. So, near optimum solution is sometime called a local optima, and then guaranteed optimum solution is called the global optimum.

So, there are minima as I told you and then out of this minimum, the global minimum is the guaranteed solution, but sometimes genetic algorithm can converse into a local optima, and that local optima may be sufficient to find the optimum solution, which cannot be solved using traditional approach to find this one. So, the genetic algorithm essential does not give you the correct solution always, but is a near correct solution actually.

(Refer Slide Time: 07:02)



Framework of GA: A detail view

Now this is the concept of the genetic algorithm that is followed their, and I just want to detail as it the different steps involved, namely the solution generation, and then the selection operation, the reproduction operation, and all this thing that is there in this framework.

So, we have to start it, whenever we start with; that means, we have certain random solution at our hand. So, the random solution basically gives the initial population, now in order to have this random solution, we have to have full idea about the different parameters which are involved in this optimization problem, and then all these parameters are to be taken into account and this should be represented in some form for each the GA framework and handle them so, it is called the parameter representation, once the parameters are identified and then parameters are represented precisely, then with the help of this representation we can generate the population, population means random solution.

So, initially if we generate a population such a population is called initial population. Once the initial population is observed we have to check about whether the initial population already got the optimum result or not; that means, the best solution or not, if it is yes then definitely we can stop this process; that means, our work is done, if it is not then we have to go to the next population generation. Now so far, the next population generation is concerned it basically needs 2 tasks, these tasks are relative selection and in reproduction. Now so far, these 2 tasks is concerned it basically evaluate the fitness of a solution; that means, how solution is good so, that is basically evaluation and now whenever this evolution is there so, it basically considered some cost function, we will learn about that how to evaluate a solution or how to evaluate an individual in the population so, that it fitness value can be calculated.

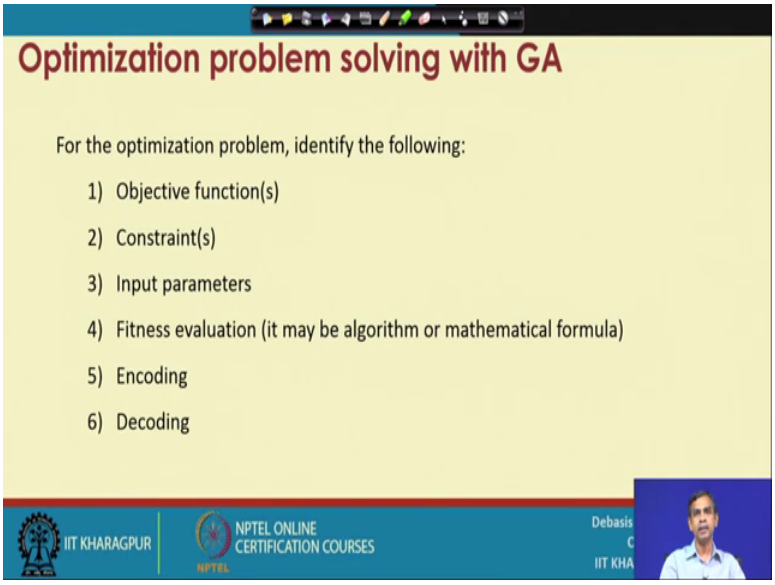
So, these are the theory are there we will follow the theory, and then based on the evaluate we have to select mating pool; that means, out of the individual who can be responsible for the mating process so, that they can produce next offspring. So, that there is a select mating there is an again there is a theorem by which from this fitness evaluation value we can select the mate. Once the mating is done; that means, the mating pool is created; that means, this individual will mate with other individual and so on, so on. So, this mating pool will undergo reproduction scheme now reproduction is a there are some few steps the reproduction crossover, reproduction by means of mutation, reproduction by means of inversion.

So, this reproduction in fact, produced from a mating pair to another individual. So, successfully there are set of mating pairs from these mating pairs, we can obtain one or more results or it is called the offspring. So, these offspring produced the next generation. So, next generation will be here again the next generation will be tested, whether the next generation achieved the best result or not, if pairs will stop it here if no we can repeat the same procedure, until this convergence criteria is successful.

So, in genetic algorithm few things are important how to create a new population? And then how the selection can be done? Selection by means of evaluation and then mating pool generation, and once it is there then how to go do the reproduction, then means how the crossover mutation and inversion can be carried out so, that next population can be obtained, and for each population we have to check the convergence.

So, this is the one important tasks, this is second, this is third and finally, convergence. So, learning a genetic algorithm in fact, learning these 4 tasks in details. So, we will learn all these 4 tasks in details and then finally, we will see how given an optimization problem, we can solve this problem using genetic algorithm.

(Refer Slide Time: 11:40)



Optimization problem solving with GA

For the optimization problem, identify the following:

1) Objective function(s)
2) Constraint(s)
3) Input parameters
4) Fitness evaluation (it may be algorithm or mathematical formula)
5) Encoding
6) Decoding

Now, will discuss about few things here, for the optimization problem, we have to consider few things, first given an optimization problem means we have given an objective function, objective function in fact, defined in terms of some input parameters, basically with the these are the parameters whose values decides the value of the objective function. And as I mentioned already an optimization problem is specified by means of objective function, and then a constant set of constants.

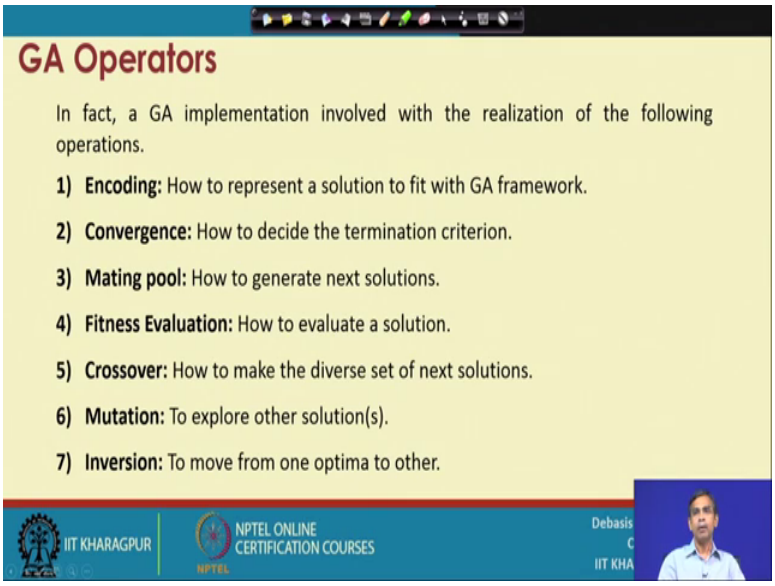
The constants are basically the requirement by which all these values should satisfy so, that the optimization function will get it is value. And so, input parameters are involved there so, input parameters are basically the input values to the system, and then the fitness evaluation and for every solution we have to calculate some fitness value; that means, if say suppose solution is the, optimum solution one or global solution one, then it should have the highest fitness value, if it is very far from the global optimum then it fitness value is also very far from the optimum fitness values.

Now in order to represents a solution in the form of a genetic algorithms individual, then we have to follow encoding, encoding means how the parameters can be represented so, that it leads to a chromosome. Chromosome is the basic concept that is there in the genetic.

So, encoding is nothing but representation of chromosome for a solution, and one solution we can consider an individual. So, chromosome is basically decide an individual them. Now chromosome is basically encoded form; that means some symbolic representation. So, we have to follow decoding so, that from this symbolic representation we can get the actual values. In other words, a parameters that is there we have to encode it so, that the chromosome can be represented this chromosome defined the individuals proper solution or population, and for many solutions the many chromosomes and then many what is called the individuals, and then population is generation.

Now encoding is basically a process by which a chromosome can be obtained for a given problems or in terms of input parameters, how the encoding can be done so, that input parameters can be converted into the encoded form, and decoding is basically the reverse form of the encoding now.

(Refer Slide Time: 14:33)



GA Operators

In fact, a GA implementation involved with the realization of the following operations.

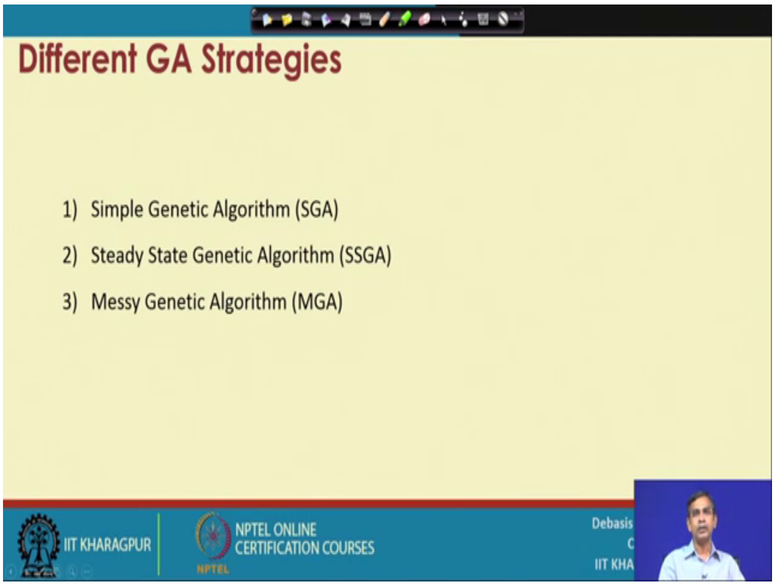
1) **Encoding:** How to represent a solution to fit with GA framework.
2) **Convergence:** How to decide the termination criterion.
3) **Mating pool:** How to generate next solutions.
4) **Fitness Evaluation:** How to evaluate a solution.
5) **Crossover:** How to make the diverse set of next solutions.
6) **Mutation:** To explore other solution(s).
7) **Inversion:** To move from one optima to other.

So, these are the optimization problem solving approach with g a, and so, far the GA operation is concerned, those are the operation that we have mentioned is carried out by means of called the operators, those operators are basically the functions like.

So, for example, encoding convergence, mating pool, fitness evolution, crossover, mutation inversion these are called operators. These are basically the functions like so, if we give an input this function will produce an output; that means, if encoding is concerned it basically represents a solution or that chromosome for a given input parameter or set of parameters. Now convergence is basically if we get a population or a generation, it will check whether we have reached to the termination condition or not, then you have to stop it or you have to continue the process of the genetic algorithm. Then mating pool is basically given a population we have to create the mating pool. So, mating pool is an function or it is an operator, by which input is a population and it produced the output as the pools, mating pool similarly fitness evolution it is also a function input to this function is basically an individual it will return a result giving the fitness value of that individual or solution.

Crossover is basically if we pass 2 chromosomes to this operator, it will produce 2 or more offspring. So, basically the crossover is an operator, inversion is basically to jump from one optimum value to another optimum value, how it can be done. So, if we give a population here it will produce another new population that can leads to a better optimum value like. So, these are the operators so, learning GA is basically learning all these operators. So, we will discuss about how these operators can be realized in terms of some simple problem.

(Refer Slide Time: 16:33)

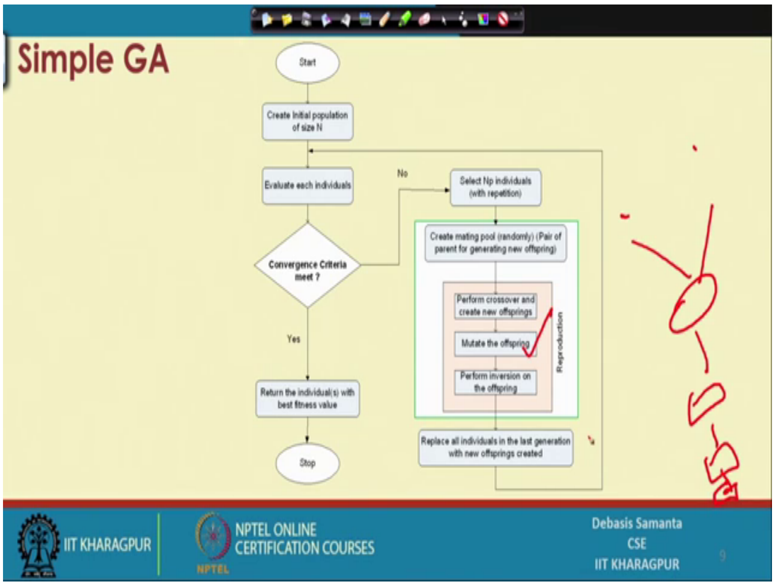


Different GA Strategies

1) Simple Genetic Algorithm (SGA)
2) Steady State Genetic Algorithm (SSGA)
3) Messy Genetic Algorithm (MGA)

Now, before going to actual learning about the different operators, there are few what is called the strategies for the g a, one is called simple GA it is also called simple genetic algorithm, and another is called the steady state GA or steady genetic the algorithm, and alternative to these 2 GA strategies there is also one called messy genetic algorithm. In this lecture we will only discuss our we limit our discussion to the first 2 than SGA and SSGA, simple genetic algorithm and steady state genetic algorithm these are the most widely followed strategies. So, for the genetic algorithm is concerned once we learn these 2 strategies will be able to learn the genetic algorithm later on who in terms of their different operators

(Refer Slide Time: 17:20)



Simple GA

Start
Create Initial population of size N
Evaluate each individuals
Convergence Criteria meet ?
No
Yes
Select Np individuals (with repetition)
Create mating pool (randomly) (Pair of parent for generating new offspring)
Perform crossover and create new offsprings
Mutate the offspring
Perform inversion on the offspring
Reproduction
Replace all individuals in the last generation with new offsprings created
Return the individual(s) with best fitness value
Stop

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Debasis Samanta CSE IIT KHARAGPUR

So, let us first start with the simple genetic algorithm, it is called the SGA and this architecture this block diagram or flow chart, shows how the simple genetic algorithm works. We may note that this simple genetic algorithm is basically a little bit detailization of the genetic algorithm framework that we have already learned now; that means, the different researchers followed the different way to solve the genetic algorithm, they just do certain different follow certain different strategies.

So, we will discuss about the pro strategy that is followed, they are in simple genetic algorithm now we will just step by step here. So, first as the start is; obviously, the starting point of any process, and then the initial population creation now here one parameter called the GA parameter first followed. So, initial population means it is a collection of solutions so, random solutions now whenever we say the collection of random solution then what is the size. So, it is a programmer who decides the size, let N be the number, N may be some times 100, some user may follow N as a thousand so; obviously, if we follow large values of N may be that we can come into the quick the solution or better solution, but at the cost of more timing.

So, if we it is take the population size less; that means, value of N is small then we may come quickly terminate the solution, but it may not give you the correct result always. So, as a theta of course, there, in this strategy population size is an important parameters so, simple GA, GA consider a good value of n, the N is the first GA parameter the genetic algorithm parameter.

Now once the population e population's size is decided, and following some procedure of creating the random solution as the initial population, the next task in according the simple GA is basically evaluate each individual. So, it is called the evaluation that evolution means the fitness evaluation. So, it basically for each an individual solution in this current population, the operator or function evaluate the fitness value. So, that is called the evolution.

So, this is a second step and then after the evolution of individual solutions have known, it basically come to the convergence criteria checking. For each individual it will check that, whether see suppose we know that the highest fitness value is possible for a solution is this one, if any individual have this highest fitness value then we can say that we can achieve the convergence criteria so, we can stop it there. Now if convergence criteria satisfy we can say yes and return the individuals with the base fitness value and that is the solution.

So, these basically the solution now if this convergence criteria is not satisfied, then we have to go to the next step, the next step is basically select N p the another parameter in simple g a, N p denotes the subset of this value of n, N maybe say 500, and N p maybe say 20 percent of 500, maybe say 100 so, select 100 individuals from these individuals. So, this is subset selection, and these basically allow with repetition.

Now whenever it is repetition we can select here basically how we can select. So, you the repetition is allowed; that means, we can select at random. So, we select one returned to this individual again, select returned the same individual and so on so, if we proceed if we follow the same procedure for N p times. So, it will gives N p individual selection, from this population with repetition.

So, this is the idea and we can select it at a random fashion, randomly we can choose one and then select and then being into the in this pool then next again this one and so on so on. So, N p individuals with this selected, and with repetition; that means, same individual may be selected one or more times here. So, this algorithm allows this one to select one individual more than one type.

Now this select again there are certain heuristics or some principle or policy to be followed. We will discuss about what the policy that needs to be followed in order to realize the selection operation. Next once the N p individuals a subset of this set is selected, our next procedure is this is basically the selection algorithm that is there. So, selection operation and I told you that selection operation is based on the concept of evolution, which basically has the 4 premises heredity, diversity, and then ranking and then selection, anyway those things will be discussed again I will discuss the selection procedure in details, and next whenever the select N p individuals are selected there create mating pool, and here again mating pool should be created randomly. So, these are the different individual if it is they are.

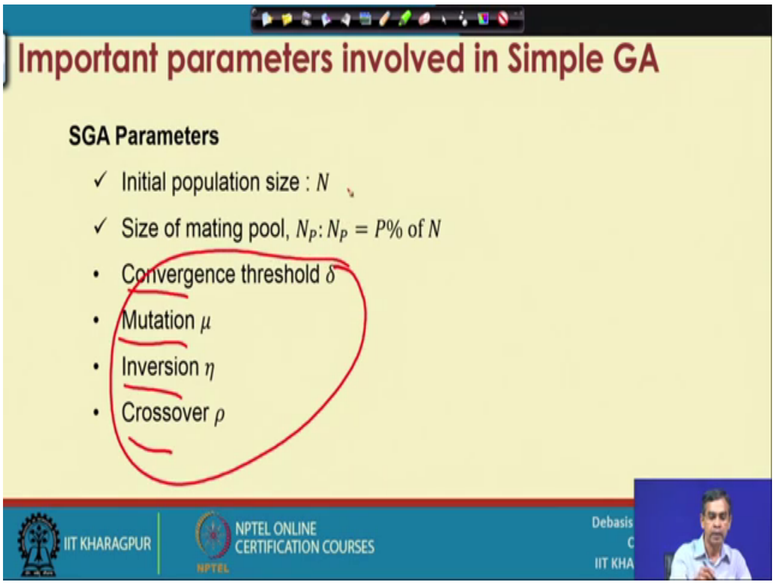
So, randomly this one or this, one mating pool. So, this one and this one are mating pool, this one this one another mating pool, this one this one mating pool. So, this way you can randomly select certain pairs, and these pairs gives you the mating pool. So, you can create mating pool is basically random process. So, this is also random process, this is also random process, here also initial population generation random process, that is genetic algorithm we call the probabilistic search, random search.

Now once the random randomly those mating pool is selected then for each individual there is a chromosome so, from this they are haploid and then from a haploid. So, cross over and then diploid it will be created. So, this basically concept the crossover it is haploid here and in this algorithm and then reproduction is there, and mutation is basically is a another approach, it is also in nature say that all of a sudden sub chromosome is there, there is a some breaking in the code or d N a code, like or genetic code. So, all of a sudden drastic change or some changes in the genetic code, it basically called the mutation, and in version is basically I mean if mutation is a very minor changes where in version is a very detailed changes. So, the detail changes is called inversion.

So, so what will happen is. So, if the 2 i mean chromosomes from their offspring is created, this offspring undergo mutation, it is not necessary mutation should go or not go with certain probability, and then inversion then finally, it will give you the new offspring. So, this new offspring will be stored into the new population or new generation, and then all these new generation that has been obtained, then these basically out of these N p. So, certain number of new offspring will be generated.

So, here according this simple GA replace all these new individual that obtained, in the last generation with new offspring created. So, we have selected out of in N p those N p will be replaced by the new offspring. So, it will remain the same size. So, here basically N and then N p. So, these N p will produce new N p, and then new N p will be replaced by the old N p so, the population size will be N and then new population the next generation will be there. So, this is the idea of the simple g a.

(Refer Slide Time: 25:28)



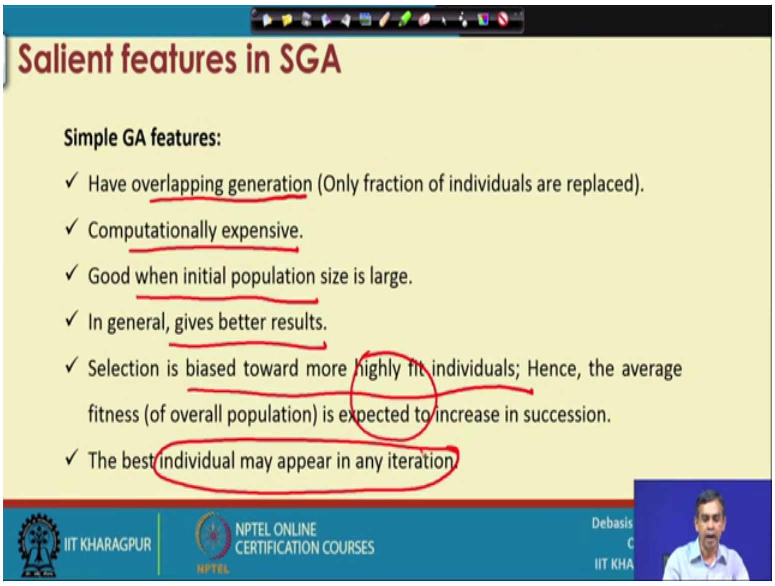
Important parameters involved in Simple GA

SGA Parameters

- ✓ Initial population size : N
- ✓ Size of mating pool, N_P: $N_P = P\%$ of N
- Convergence threshold δ
- Mutation μ
- Inversion η
- Crossover ρ

And so, next idea is other than simple GA it is called the so far, the simple GA is concerned few things are important. So, as I told you in the simple GA few parameters N these are initial size of the population N p is basically, size of the mating pool it basically p percentage of n, p may be some value decided by the programmer, all these N p these are the value decided by the programmer, who will use these genetic algorithm to solve their problem. And other than these parameters few other parameters are also involved these are the cry of a convergent threshold; that means, it is basically the range by which we can take so, that our result is near optimal or near ah minimum or maximum, and then few other parameters are there so, these are called mutation parameters inversion parameters and crossing over parameters. So, we will discuss about all these parameters when we will discuss all these operation one by one. So, till time we can keep it on hold.

(Refer Slide Time: 26:25)



Salient features in SGA

Simple GA features:

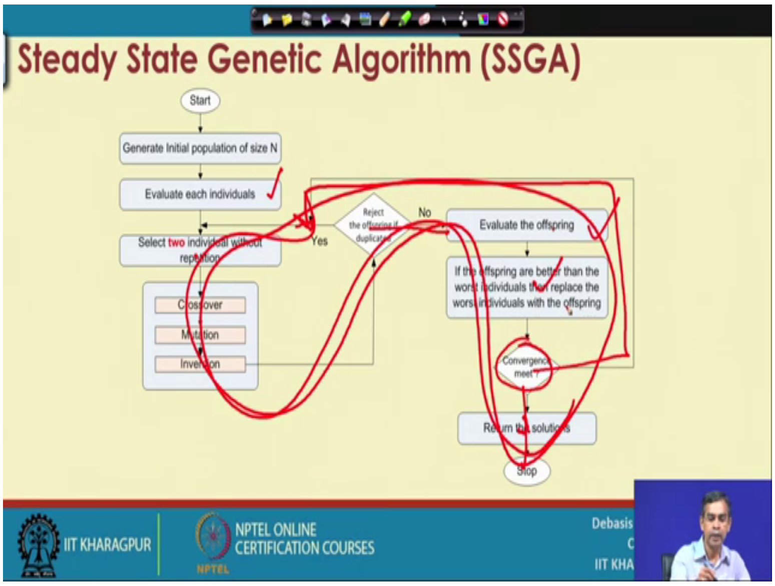
- ✓ Have overlapping generation (Only fraction of individuals are replaced).
- ✓ Computationally expensive.
- ✓ Good when initial population size is large.
- ✓ In general, gives better results.
- ✓ Selection is biased toward more highly fit individuals; Hence, the average fitness (of overall population) is expected to increase in succession.
- ✓ The best individual may appear in any iteration.

Now, let us see the other algorithm that is there. So, to discuss about it now there are few important features in SGA money in which situation we should follow the simple genetic algorithm so, simple genetic algorithm always produce overlapping generation; that means, only fraction of individuals are replaced; that means, from the current generation to the next generation, few individuals are remains common. So, it is called the overlapping generation. So, these the one characteristics because, some solutions are common in between the two-successive generation.

And this simple GA is usually compression expressing compared to the other GA strategies, and this kind of GA is applicable, when initial population size is large, this general usually gives better results compared to the other strategies, and here in this algorithm selection is biased towards the more highly fit individuals; that means, it basically selects those individual which are having the highest fitness value, but it is observed that sometimes selecting, non-highest fit individual value and then taking the crossover can leads to a better solution. So, this has the one what is called the issues ad here. So, that highly fit individual selection not always useful rather, if we select some other individual having the lower fitness value may give the better result or may give quick result.

Anyway so, these are the another thing and these are the best individual may appear in any iteration; that means, it can give the best result in any iteration, and then that can be one achievement or that good point so, that we can terminate quickly if you are lucky enough then with one iteration maybe you can terminate, but if you are not then you have to repeat it again, but the chance that it will terminate very quickly higher compared to the other strategies.

(Refer Slide Time: 28:30).



Steady State Genetic Algorithm (SSGA)

Start
Generate Initial population of size N
Evaluate each individuals
Select two individual without repetition
Crossover
Mutation
Inversion
Reject the offspring if duplicate
Yes
No
Evaluate the offspring
If the offspring are better than the worst individuals then replace the worst individuals with the offspring
Convergence meet ?
Return the solutions
Stop
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Debasis
IIT KHA

Now so far, the other strategy is concerned, we will discuss about next strategy and then we will decide we can understand about what are the difference between s GA versus this strategy, now the next strategy is called steady state genetic algorithm. Now let us understand what are the strategy it is, as in the simple genetic algorithm it will start with the initial population size n, N is decided by the programmer evaluate each individual that is the fitness evaluation. Now here there are actually N p number of selection in case of s g a, but this s s g only select 2 individuals, now these 2 individuals one selected they then go for the reproduction; that means, we do not have to do the mating pool creation here unlike in case of SGA known it.

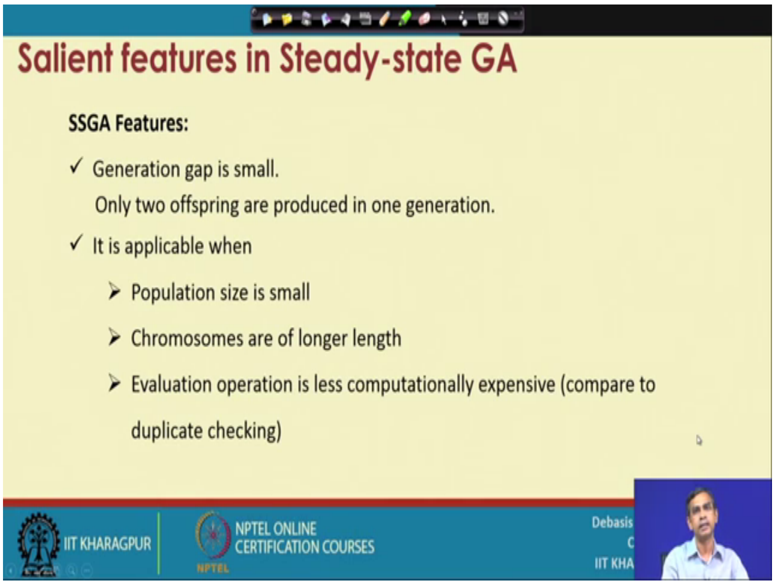
So, from the 2 individuals will be selected from the current population and then they will produce the offspring. Now here again go to that reject the offspring it is duplicate, now the offspring that will be produced if we see it is already there in this population, then we should reject this offspring then we can repeat another individual selection from here and then another offspring can be created. So, if it creates a new which is not repeated or not available already there then we can go if it is no. So, evaluate the offspring; that means, the fitness value of the offspring will be created, but in this case only one evolution is required, now here after the fitness value is created if the offspring are better than the worst individual.

So, in this population there are few individuals whose fitness value is worst. So, if the current offspring fitness value is improved value, comfort that is the fitness value is greater than the any worst individuals fitness value, then we should replace these worst individuals by these new individuals new offspring; that means, it in each iteration it will replace one offspring with worth witness value by a better witness value.

After replacing it will produce the next generation, but in the next generation you can see only one offspring is different anyway. So, if the next generation satisfy the convergence criteria, then we can stop and then these are the solutions. So, here you can see the solution having the highest fitness value is the ultimate solution if it is it does not satisfy the convergence criteria you can go it, again proceed the same procedure select 2 individuals. So, it is basically the root so, the iteration will be there.

Now we can understand the difference between SGA and s s g a. SSGA always selects in each iteration or changes the population with a larger gap; that means, gap is at least by N p, but in this case the gap is very small, only the 2 successive population is differed by means of only one two solutions.

(Refer Slide Time: 31:41)



Salient features in Steady-state GA

SSGA Features:

✓ Generation gap is small.
Only two offspring are produced in one generation.

✓ It is applicable when

- Population size is small
- Chromosomes are of longer length
- Evaluation operation is less computationally expensive (compare to duplicate checking)

So, this is the fundamental difference that we can easily understand, and here is the some features so, far the SSGA is concerned as I told you the generation gap is small only 2 offspring's are produced in one generation, it is applicable usually when the population size is small, chromosomes are of longer weak, and evolution operation is less computationally expensive; that means, when the evolution is too much expensive to calculate the fitness value of an individual, then we can follow it and chromosome length is basically if the solution needs a large number of parameters then usually we can follow this technique.

So, we have learn about s s g a, and limitation in SSGA is that compared to the SGA of course, there is a chance of stuck at local optima if crossover mutation inversion is not followed properly, premature convergence usually occurs which is not occur generally in case of s g a, and it is susceptible stagnation; that means, inferiors are neglected or removed and keeps making more trials for very long period of time. So, sometimes it is observed that if the inferior individuals are taking into the mating, they can lead to the better solution so, but it is ignored here. So, it may come to a stagnation situation.

Okay. So, we have learned about SGA and then s s g a, the 2 strategies and based on these 2 strategies how the genetic algorithm can work, and then in order to understand the working of genetic algorithm, we have to study about the different operators. So, in our subsequent lectures we will learn about the different operators that is here, namely encoding the crossover mutation the selection fitness evaluation and all these things ok.

Thank you.