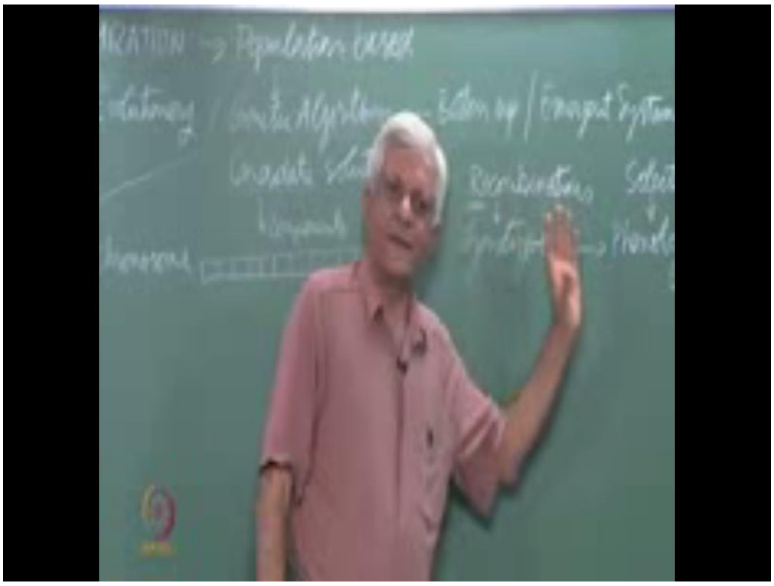
Artificial Intelligence
Prof. Deepak Khemani
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture - 16
Population Based Methods for Optimization

(Refer Slide Time: 00:18)



So, we continue looking at optimization and we are looking at methods which we called as population based. So, basically when we say population based methods and we will spend some time on this today. We mean that instead of working with a single candidate solution for example, as we do in hill climbing or in simulated annealing we work with a population of candidates. Now, the benefits that we get by working with populations I have not just speed up, because it is not as if you are running the same algorithm k times or n times if n is the size of population. You also get benefit from the fact that different candidates can contribute jointly to exploring new solutions. So, essentially we are exploring the solutions space which for real problems may not be a smooth surface. And we are looking for methods which would be robust which even when the terrain is jagged we will do reasonably well essentially

So, in population based methods the first method that we are looking at this moment we have called it is genetic algorithms some people also call them as evolutionary algorithms. And they have a flavor of bottom up problem solving we we also called as emergent. So, by emergent systems we mean that we put together a collection of simple elements and more complex behavior emerges out of that. And as we might have said before the human brain is a perfect example of that every neuron in our brain is computationally a simple device, but it just that we have billions of them in our head. And they are it is a connections it is the way that they are connected to each other that really give raise to our individual brains essentially. So, the idea in genetic algorithms is that we have candidate solutions rather a population of candidate solutions and each candidate is made up of components.

So, you can think of this some kind of a breakup of every component now in the simple case that we saw about solving sat every component is the value of one variable or one bit. And a sats can sat solution candidate is a bit vector, but in general problems may be more complex. And components may themselves very more complex in nature essentially in which case of course; we have to be very careful. Because the general strategy in solution space search or perturbation search that we are exploring is that we perturb candidates to produce new candidates. So, if we had this as a candidate then simulated annealing may try to generate a neighborhood around this by changing one or 2 components as we saw in the case of sat and do that. So, one has to do the little more carefully essentially.

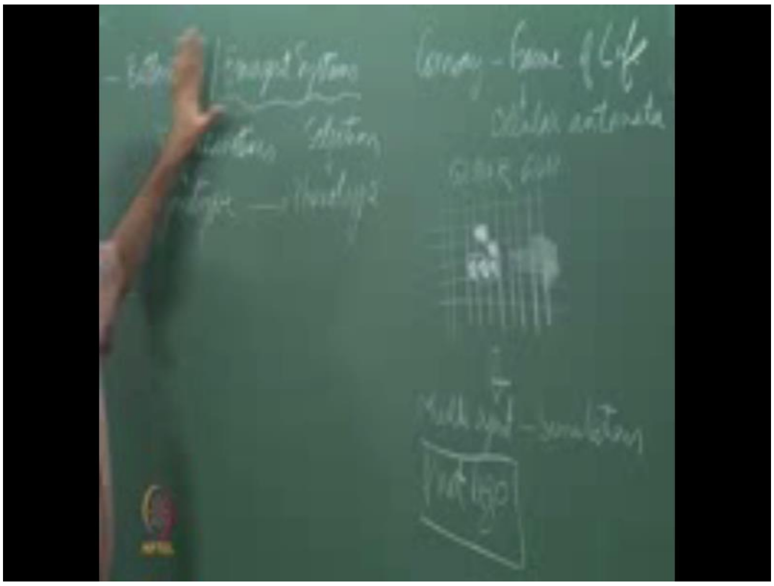
So, when we talk about genetic algorithms we have to talk about encoding candidate solution as a string in terms of genetic algorithm. We would call this as a chromosome borrowing from the biological world. And essentially there is an interplay between chromosomes of different parents essentially which basically modifies the design of the candidate or design of the solution essentially. So, we had mentioned in the last class that there are these 2 things; one is the genotype which is made up of the genes. And that in that influences or they decides the behavior of the phenotype which is the individual or the creature or whatever is participating in the real world essentially. So, in genotype we look at recombination’s and the phenotype basically competes and its participates in selection. So, these are the main 2 components of the problem solving strategy that you might say nature employ employs which is the trial and error with its designs.

And the design is expressed as a chromosome essentially which is some kind of a string that we are going to work with and we said the use a word recombination’s. So, for idea in genetic algorithms is that you draw some components, some one parent some components on another parent and in the hope that you will inherit the good components of both parents. So, if you going back to the rabbit and foxes example, let us say there are 2 desirable properties amongst rabbits. One is that they can run fast and other is that they are smart so whatever that means you know. So, for example, you must have as a child heard a story about the rabbit who has to go to the lion in the jungle as food. And then he is a smart rabbit and he tells a lion of another lion and that kind of stuff. So, either they are fast or they are smart essentially which is contribute today's survival.

So, the whole idea of evolution is that in a population of candidates, the phenotypes, the fitter ones are the ones who survive the competition. Competition is for resources it could be for food, it could be for mates, it could be for shelter which is you know safety from the predator essentially. So, if you are good at doing these things then you are likely to survive which means you are likely to pass on your genes to the next generation and so on and so forth. Now, imagine 2 parents of a rabbit; one of them is smart and the other one is fast essentially. The hope in genetic algorithm is that if these 2 parents have let say 2 children that is module that we are following here at least one of them will inherit. Both these properties of being smart and being fast at the same time which means it will be like a super rabbit essentially right and super rabbits will give rise to more super rabbits so on.

So, in general the rabbit population will become smarter and faster essentially. So, that is the whole idea behind genetic algorithms like we said one makes up the combinations. So, there is one processor one force which makes up the combinations jumps the genetic pool and the other which is the process of selection which is that they compete for survival essentially. So, survival of the fittest and when we say survival of the fittest at least in the case of genetic algorithms we are going to impose a fitness function on top of the algorithm. Because it is based on the fitness function that we will the candidate will survive, but in the real world its other way round it is basically whoever survive this is the one whose fit essentially. So, I want to do one example of to illustrate genetic algorithms today, but before that I want to sort of spend few minutes on this idea of emergent systems essentially. Because it is an idea which is has a lot of follows essentially.

(Refer Slide Time: 09:32)



So, you must have heard about this conveys game of life convey something called game of life. Have you heard about the game of life? It is like a screen saver sometimes and the world is a grid in cellular voltammeter. So, you know we have rows and columns and so on and in the simplest form of this game in every cell. So, every square is a cell inside there can be one creature and creature is a very simple creature it survives. So, it has some rules for survival there are rules for surviving there are rules for being born even in this very simple world there rules for dying essentially. So, the simple rules in this conveys game of life are that if you are surrounded by more than three living cells or creatures in living in cells. Then you are sort of overcrowded and you die essentially which means that each cell is in 1 or 0 1 means its alive and 0 means its dead or the creature inside is dead.

So, if there is a one surrounded by more than three ones then it dies it becomes 0 if a 0 is surrounded by exactly three ones then it comes alive. And if a 1 is surrounded by less than three ones and it stays alive essentially or something I may have a small error there but anyway something of that nature. So, you have this, very simple rules which control how a cell evolves over time. So, it is an automate which goes through a sequence of time steps. So, at every time step every cell decides whether in the next time step it will be alive or dead or whatever essentially. Now, it turns out that this very simple rules of life if you want to call it life give rise to patterns in this world which is a grid world and which have a tendency to survive. And some patterns survives some patters oscillate between 2 3 patterns and some patterns even move forward.

So, if I remember correctly there is a pattern which looks like this that if this is one and this is one and this is one. So, these five patterns if they are one and maybe you should try the rules that I just mention on this, these cells. Then this pattern essentially in the next cycle we will move little bit to the right may be it will rotate by a little bit and it will keep doing that. So, as the time progresses this pattern will appear to be moving in this direction. Now, this as sort of takes me back a little bit to the introduction that when you were looking at you know what is reality? What is really out there? And that kind of stuff we were discussing in this simple world. What is really out there is that these are the cells which become 1 or 0 essentially. And that is about it this whole world is full of cells and each cell can be 1 or 0. And they follow certain rules whether to remain 1 or 0 in the next time cycle.

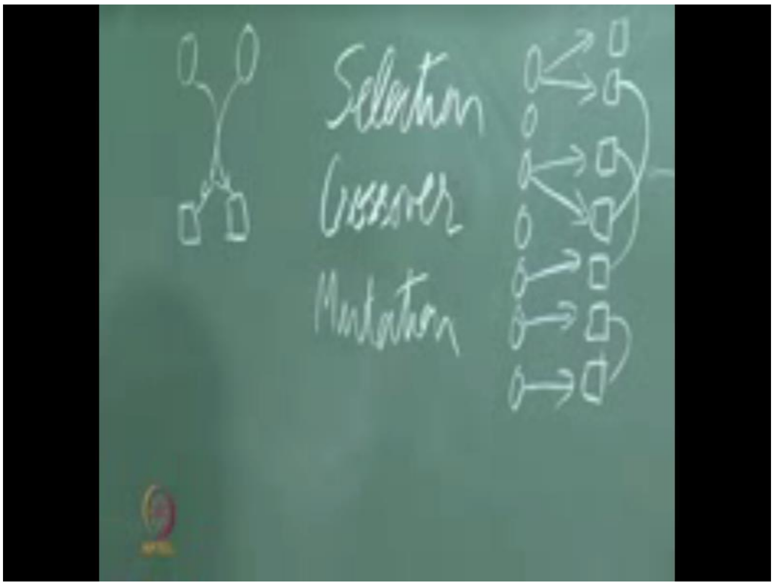
But in this world certainly we see patterns like this; this is called the glider gun and this is the very old game, game of life was invented in 60's or something like that essentially or may be 70's. So, if you just search game of life or glider gun on the web you will probably get to see on animation of this. So, this looks like a creature norm you know of this shape which is moving around essentially. Of course, we are also like that in some sense made up of how many 10 rise to something I had mentioned 28 or something simple cells moving around in a unison pretending to be a creature. But that is a whole idea of emergent system essentially that you put together simple cells or simple components and put which obey certain local rules each cell only looks at it only neighborhood. It does not see anything beyond its immediate neighborhood and that it its fate is decided base based on that essentially you put in such local rules and build a world like this.

And you will find that patterns I mean because we tend to think of them is pattern emerge essentially. So, basically combinations of simple things tend to behave in a more complex way essentially. So, more recently there has been a lot of interest and now imagine that instead of being this very simple creatures that these cells are which only look at the neighborhood. And decide whether to die or not to die they were slightly more complex creatures. For example, they could have a little bit of a memory of their own and they could have a little bit of listening something. Then we sort of call them as populations of agents or some people call them as multi-agent and you are talking about simulations here. So, a lot of people for example, social scientist are looking at these multi-agent worlds in which they put together this simple agents and they study the behavior of those. Of course, they put together the rules and so on so forth for example, work ordering rules or things like that and sort of look at more sophisticated implementations of this.

So, I would there is a language called net logo so you must have heard about logo of course, devise by paper long time ago to teach programming to children. So, there is a total which you can say move north or move west and so on so many step certain by so many steps in that kind of thing its net logo is the more sophisticated language. It has been in implemented by some people in the north western university in the US and its available freely online and many. And I have some friends who are sort of working with this and trying to implement these agent based simulations. And so this whole idea of exploration is to put together simple elements impose rules upon them and then study the behavior. So, people have studied for example, crowd behaviors and one of my friends who works in TCS in Pune. He is been doing this sort of a thing and he is sort of simulated this behavior of this crowd.

So, there was this fire in some restaurant in Brazil or some are in the crowd behavior he could replicate that crowd behavior by imposing very simple rules on the agents essentially. So, what do you do if you are in a restaurant or a bar and there is a fire in one corner of the room? What would you do essentially you know, so impose simple rules upon then and then you can actually study the behavior of crowd essentially. And nowadays of course, crowd behavior is of great interest specially in situations like you know this Uttarakhand tragedy or any tragedy that befalls upon this. You want to see how to how the crowd would behave essentially. So, the so there is one direction to this population based system is to is this agent based simulation you create simple agents put them together in a world. And then simulate or run the world to see how what kind of behavior would emerge essentially.

(Refer Slide Time: 18:25)

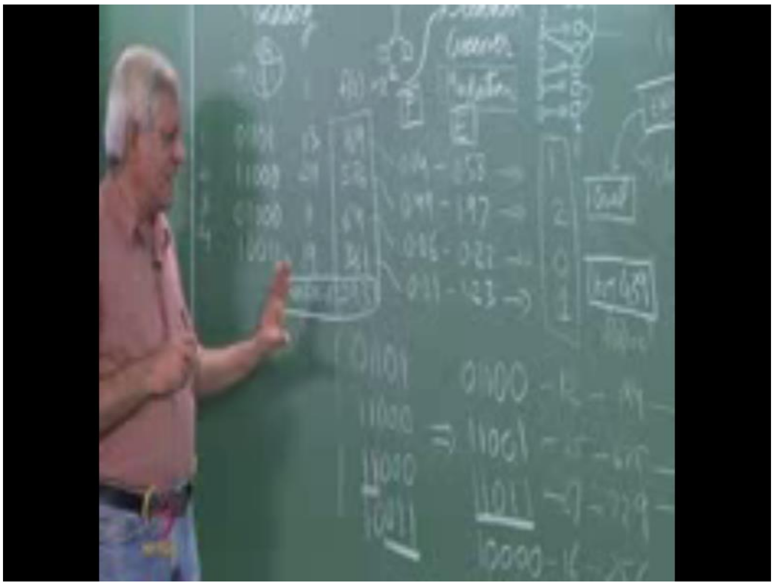


But we will come back to our genetic algorithms which is also little bit like that except that there is no spatial connotation here we have a space which is made up of grids and cells in which the people live but in g a’s we just have a population. And we do not really talk about space essentially though there have been people who have also started speaking about spatial distribution of populations essentially. So, just as human populations are sort of distributed in different continents and different countries and so on and so forth. They have try to see whether you know the whole population can be partitioned into some sub populations and evolution basically interaction mating. For example, happens only within that population and not outside this population. And then they try to impose rules like migration when is the agent allow to migrate to a new country. If I am use the word under what conditions will this migration be successful under what condition will it fail and that kind of thing essentially. For, so those of you have been following Australian news and elections are coming up they have been taking a very strict stand against emigration.

So, they have said that anybody who arise by boat will never be given Australian’s this is in ship essentially you will also rules like this. And you can see how it will affect the population one can think that but we work with simpler notions which is that we just have a collection of populations and what is the algorithm that we had talked about the first step is selection. So, we have this population of some elements and they are allowed to reproduce themselves based on some fitness value something like this that is the first step which is selection. And we had said that we will implement the rule wheel kind of a mechanism of course, you have to work out how to do that which will tell which element gets to reproduce. So, if there are n elements we will spin the we will n times and whichever elements comes in front of let us say the pointer we will get to reproduce essentially then crossover which basically means that we randomly pair the new population that we have. So, let me draw this by squares we randomly pair them. So, for example, we could say we pair this one with this one, this one with this one, this one with this one maybe I should have an even number.

But still we do random pairings and allow the cross over operator which is basically mixing up the genes of the 2 parents and producing 2 new children. So, you can centerline sort of see this as a paddle search in the same space solution space except that we start with a move is made up by somehow selecting 2 parents. And then generating 2 new children out of it instead of one parent giving rise to one child like for example, hill climbing or simulated annealing here 2 parents give rise to 2 children in this example essentially. And then we had said mutation which is kind of a rare event once in a while you change some gene randomly essentially. This is meant to take care of the fact that some genes may get lost in so on and so forth essentially. So, let me take an example to sort of illustrate how genetic algorithms work and then we will come back to discussing what are the issues that you have to decide upon when you are using genetic algorithms for optimization So, this is an example which I am taking from this book by Goldberg.

(Refer Slide Time: 23:00)



So, remember I had told you that the genetic algorithms are device by somebody called Holland John Holland and his student called David Goldberg has written this very popular book on genetic algorithms. And this is basically from the first chapter which is the introduction chapter of his book. And it is a nice example and I like it, because it illustrate the whole idea of g a quite nicely. So, it is a very it is called very small population of size 4 so we have this 4 elements. So, remember that the first thing we want to do is to encode this. So, when we want to use the g a to solve some optimization problem we have to first encode the candidate solution. Then we have to devise an evaluation function which will look a candidate and give us a number. And ideally the evaluation function should be in sync with what we really want to do which means that for the optimal solution the evaluation function must have the highest value essentially.

So, let us assume that we will do that and then with the genetic algorithm thing comes into play. Now, in this very simple world I have a 5 bit vector and so the 4 elements that I have in this example. So, I have a small population of size 4 and these are the 4 candidates represented as bit strings. And let us say that we implement in we interpret them as binary numbers which means that the numbers that we are talking about are this is 13 this is 24; this is 8 and this is 19. So, essentially these are the candidates but this is the representation. So, we have this 4 numbers 13 24 8 and 19 and we want to know our optimization task is to find the largest number that you can represent with 5 bits. It may sound like trigger to you, but we want to see how genetic algorithms work essentially and that is our goal and it is a problem that we understand quite clearly essentially.

And, so we have to have this evaluation function right. So, let us say we decide so this f is a fitness function which will take any candidate and give us a value. And just simply let us assume it is a x square essentially a simple fitness function essentially. So, if you want to evaluate this 4 candidates the values that we will get 169 596 64 361. Now, what is the role of fitness in our algorithm? Our fitness influences the selection process the chances of a candidate reproducing itself our proportional to fitness. So, in our rule wheel we will have this 4 sector with areas proportional to these numbers essentially. So, this second candidate will have the largest area the fourth candidate will have the next largest area and so on and so forth.

So, we will have a rule wheel like which looks like this something like this. So, this is 2; this is 1; this is 3 and this is 4. So, we just rotate it and whichever thing comes here we select that essentially. So, the probability of it reproducing is proportional to the fraction that this is of the sum of this whole thing. So, this probability value is 0.14 for this. So, how did I get this values basically you sum of this 4 numbers and then this divided by the sum is 0.14 this divided by the sum is 0.49. So, that is how you get the probability values which means that every time you spin this rule wheel the second candidate has above 50 percent chance of being reproduce this candidate has about 30 percent chance. And these 2 candidates have little less chances essentially that is how the areas are divided essentially.

So, the expected values let us call them e are basically these values multiplied by 4, because we are going to spin the wheel 4 times in our case. So, the number of incenses of these candidates that we expect to see is basically 4 times this. So, this turns out to be 0.58 1.97. So, what are these numbers? These numbers are saying that after I have done this selection space. And what is the selection space I will spin this wheel 4 times and whichever candidate this arrow points 2 I will make a copy of that candidate. So, after I spend 4 times I expect to see about 2 candidates of this about half candidate of this about one fifth of this and 1.25 of this roughly is essentially. But of course, in the real world these fraction do not mean anything I in the end I will end up seeing 4 candidates. And let us assume that when we actually carryout this experiment the values that we get are 1 2 0 and 1 its quite feasible, it is quite possible that if I will spin the wheel 4 times.

I will get 2 copies of the second candidate one copy of the first one copy of the fourth and none of the third one essentially. So, let us say this is what happens this is where the random process is this is the first place where this random process is showing its effect in the process of producing the next population. So, this population is actually this the one that we have said after we have done the selection space we started with this and we end up with this essentially. So, let us redraw that population so 0 1 1 1 copy of this 2 copies of this and one copy of this. So, this is my new population so selection space is over now begins a crossover space. So, how so let us say we pair these 2 and then we pair these 2 and we put some random crossover point. And we are doing this single point crossover which means at some point we will break this chromosome into half and the same point will break the next one into half and then we will exchange the 2 essentially.

So, let us say that our crossover points are like this one is here and the other one is here. So, this will give us a new population So, this 0 1 1 0 will come here followed by a 0, because this 0 comes from the other side likewise this 1 1 0 0 will come here followed by this one which comes from the other side. And similar thing will happen here 1 1 0 1 1 and 1 0 0 0 0. So, just to emphasize I take this part and I take this part and I get this this one here and likewise for the other 4 essentially. So, this is my new population at the end of one round one cycle of this let us see whether we have a better population to start with. Now, if you look at this average fitness for this it happens to be 293. And the people who work with genetic algorithm tend to talk about average fitness of this thing that in general the population is becoming fitter. But of course, we are interested in the most fit element that is the different point essentially.

So, let us see how is the new population. So this numbers corresponds to 12 25 27 and 16 and there are f of x which is equal to x square corresponds to 144 625 729 256. So, the first thing is that average for this new population is 439. So, starting with this average of 293 we have gone through one cycle of reproduction and mating or in crossover and we have got a new population which is this. So, we started with this whose average fitness was 293 and we ended up with this whose average fitness is 439 essentially. So, this is the kind of thing which g a's are looking for that can we get fitter populations and so on essentially. So, the whole idea is that if you have some function in which you are trying to optimize. Then initially you are population may be decrease sort of spread over this whole domain where this is the let us say one dimensional domain. But after you have done the g a's sufficient number of times you will find that the population is kind of concentrated towards the peaks.

And one of them hopefully is the solution that you are looking for essentially. So, that is the general idea behind genetic algorithms. So, let us look at this next population and what would happen in the second cycle? We will not repeat it but we will just start thinking about it essentially. So, we look at the probability again if you remember the probability is obtained by taking the sum of all these 4 numbers. And if I 144 divided by the sum would give me the probability of this one showing up in this rule wheel the rule wheel has change now. So, this happens to be 0.08 this happens to be 0.35 this happens to be 0.26. So, these are the 4 probabilities value and corresponding to them the expected value which is basically 4 times this number is 0.32 1 1.4. So, these are the values so want you to now inspect this new population. So, we started with the one population whose characteristics are given here average fitness is 293. And this as the expected number of cloning's allowed here also we can see the expected number would be I mean these are the expected number sorry 1.97 0.22 1.23 and so on. And those are the expected numbers here 0.32 1.4 1.66 0.58. So I wanted to look at this new population and its fitness values and make some observations well, average fitness has gone as you can see is become 439. The maximum fitness here goes 576 and there it is 729. So, that is also gone so is there any problem? What will happen if I continue this cycle? Will I get what is the solution that I am looking for?

Student: One solution.

Five ones; so if I run this algorithm let us say for hundred cycles what is the what are the intuition say? Will I get that or not?

Student: Probably little bit is 0.

Exactly now, if you look at the expected values of these 4 elements, these are the 4 elements I have and the expected values are 0.32 for the first one 1.4, for the second one 1.66, for the third one and 0.58 for the second one. So, it is very likely that when I do this random spinning of the rule wheel and produce 4 new candidates out of 4 copies of sum 4 of these it is very likely that this first candidate is going to be left out. Why because expected values its probabilities only 0.08 and if I spin it 4 times my expected value is 0.32 essentially.

Notice that if I want to spin it 100 times its value would expected value would go up, but that what amount to saying that these 4 candidates are being clone 100 times. We are not doing that we are assuming here that the population remains constant that we start with this 4 candidates and clone 4 new candidates out of it. And if I do that based on their fitness values its very likely that the first candidate will get left out and as he pointed out this is the only candidate of these three which has the one in the middle place essentially on the third bit essentially. So, if I want to remove this, so let us say I want to take 2 copies of this, one copy of this and one copy of this then no matter what future churning that I do with this, I will never be able to get the middle bit as one. So, I like this example, because it illustrates exactly this point. Now, what is the ruminative this? Why is this happening? How can we try to see that this does not happen, go back to nature if you want to essentially

Student: The writing in the genetic the writing (refer time: 39:16)

When will they become more and more uniform? What you are saying has the point that yes it is possible that for a given species. So, let us call it as species since you are talking about nature as well. So, if you take for example, the cheetah now the cheetah if biologist want to look at it is. And if you want to call it a machine is a perfectly design machine for hunting its it can in fact, you know you get to see car adds which kind of try to portrait them as cheetahs and stuff like that. And the reason for that is that they can escalate very fast and they can attain very high speeds which is good for hunting essentially. Now, the cheetah was such a good predator such a good hunting machine that they all became similar in a sense this capacity to learn very fast and catch free came in all. So, the genetic the diversity in the cheetah population became less and less and less essentially. And if the diversity becomes less it means your tendency to explore the space decreases essentially. This is what is happening with this here if we have getting one candidate without the middle gene. Then you can never look at solutions which have the middle gene essentially.

So, likewise if you lose something's then you become rigid in your generic makeup which is happen to the cheetah. And which is one of the reasons that since human kind change the world so much that cheetahs do not have this open spaces. And which they can go and catch their pray they are on the danger of becoming extinct essentially. Simply just like this particular creature or the gene which this creature is carrying which likely to become extinct in the next round essentially. But the question that we want to ask is what lesson should we learn when we device g a's. So, when we devise genetic algorithms what are the parameters that we have to select the first? Of course, is how do we encode the solution? Encoding the solution should be done. So, encoding followed by evaluation these are the first 2 steps; how do you encode your candidate solution? And secondly how do you what is the fitness function that you devise for essentially?

Now, remember there is this fitness function will have to be computed for the entire population. So, one of the places where I was reading about genetic algorithms they said that imagine that you are trying to devise a neural network for pattern recognition. So, you want to recognize some patterns let us say hand written characters or something like that essentially. And if you have a training set of 1000 patterns given to you based on which you are training the neural network. And you are task is to identify the parameters of neural network which basically means the number of nodes and the edge grids and that kind of stuff. What is the fitness function? You could use one fitness function; you could use is that you could test the candidate. And the candidate in this case is the design of a neural network or a neural network on the thousand training candidates that you are that are given to you. And the fitness value is the number of candidates this network correctly classifies.

Now, if you want to devise such a training such a fitness function you can imagine the amount of work the generic algorithm has to do. Simply to evaluate the fitness value of a candidate our example is very simple we just compute f is equal to x square. It is easy to do it on the board, but it not a problem that is hard to solve anyway essentially. If you want to look at more difficult problems then you have to be very careful in how do you encode the candidates worst. And in the next class, we will come back to TSP, how to solve TSP's using genetic algorithms. Then how to devise a good fitness functions? The fitness functions reflect what you want to achieve, because in the way that we have built our algorithm. It is a fitness function which dominates this selection process that this process of cloning one population into the next. And the number of copies you make are dependent directly upon the fitness functions. So, the fitness function if it is good it will do a good job for you essentially, but there is still one more parameter to be decided and that is.

Student: Randomization.

No no randomization is a built in process randomization takes place here in the selection in the spinning of rule wheel and the other place. It take place is in deciding this crossover point here where do we do a crossover? May be it can be done in some more informed fashion and people have tried out all kinds of things, but we want to look at general purpose algorithm at this one.

Student: Sir

The third thing.

Student: how do you decide crossover I mean.

Sure, so I will take your question to mean how do you decide what is the crossover operator essentially? So, in this case now, if you think about a simple situation like this where the solution is the bit string. Then crossover essentially will mix up bit strings essentially and without loss of generality a single point crossover will do like any other crossover might have done. Though as I said in the last class goal bug has investigated as the, what if there is a sequences of characters which somehow provide a good feature. How can we try to see whether they stay together, but you will not get into that So, we will assume that a simple single point crossover does a job for us and this point is decided randomly at run time whether it is done here or whether it is done here this is decided randomly at run time. So, that is the second aspects of random random behavior that is happening here one is in the rule wheel spinning and the second is in deciding where the crossover happens. But looking at this example there is a third thing which would come to mind which is crucial for the success of genetic algorithm

Student: We should take care of extinction

We should take care of extinction. So, how do you do that? Somebody had said diversity how do you increase diversity?

Student: Keep the best.

No you are look at it from a wrong perspective.

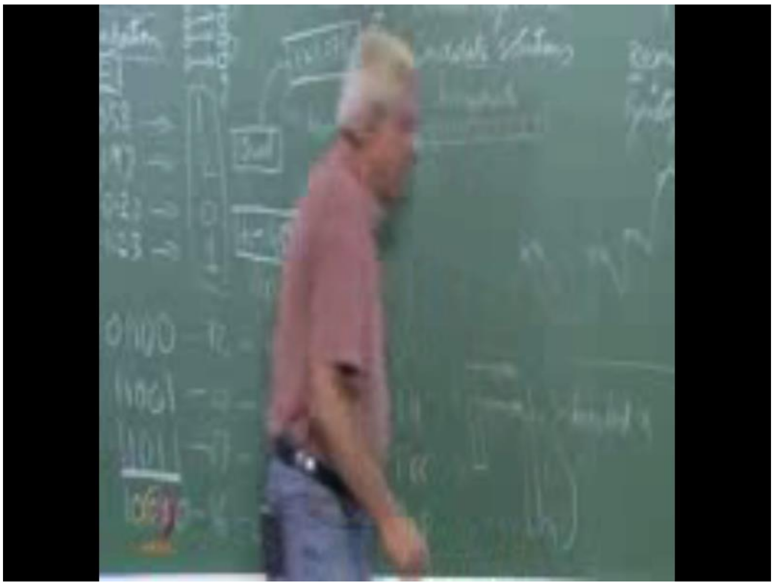
Student: All the feature if we maintained to the external.

No no listen we are trying to design a genetic algorithm; we are not trying to design the selection or the mating process right. Now, what is the third parameter in designing genetic algorithm which is going to play critical role here?

Student: Size of the population

The size of the population genetic algorithms will work if the population size is large if the population is large. Then they will have all kinds of characters inside their population with carrying all kinds of genes. And they will all have a chance to reproduce in some sense essentially. Now, which is one of the reasons why we have this third space which is mutation for the sake of completeness. There is hope that this mutation process will sometime toggle the middle bit here and we will get going essentially.. But in general you have to choose the large population size for these 2 work essentially. One more aspect that we discussed in the last class we have not mentioned it now is has to so.

(Refer Slide Time: 47:44)



So, some people have done it has follows that you have a population of some size then you take the most fit members let us say k most fit members. So, this is k this is n so if you have n members you take the most k fit members only allow those k fit members to reproduce and mix up the genes both the process selection and crossover. So, you get a new populations of k elements. Now, take this whole population here., so whatever saying that that this k elements get to reproduce which means they die in some sense in the process of reproduction that we have these 2 parents vanish from the scene and inside this new children come in essentially. So, we get this new children here k new children plus we take all these. So, this is k and this is n. So, we have k plus n elements and then you mix them up and take the best k out of them best n out of this. What we have done here is that we are not keeping the original population at all here we have said we just replace the new one.

So, you started with this population and you ended up with this population in the process we lost the no sorry this population after crossover. So, we started with this one and we ended up with this and none of them are reproduce there essentially. So, we ended up with the new population which means we replace the entire whole population with this new population. But I had said earlier also that it is not necessary that you replace the entire whole population you may want to keep some of the best ones from whole population into the new one. And this is the one approach that people take just say only allow the best people to reproduce. Then take the new children plus the old population and from that choose the best ten and that is the new population which gives a chance for good solutions to survive in this case essentially. So, I will stop here and in the next class, we want to come back to genetic algorithms and look at how to solve the traveling salesman problem. How to devise crossover operations for the travelling salesman problems and what is the representation that you can choose for representing the TSP candidate essentially? So, we will stop here.