**Artificial Intelligence**
**Prof. Masum**
**Department of Computer Science and Engineering**
**Indian Institute Technology Delhi**

**Lecture-28**
**Local Search: Local Beam Search and Genetic Algorithms Part- 7**

We have now achieved some kind of global optimality that is the point I was trying to make. By doing hill climbing with random restarts by doing hill climbing with random walk by doing simulated annealing all of these are asymptotically optimal. In the limit of infinite time you will get to the global optimum for whatever reason different reasons. Now let us build this algorithm further not this but this general space of algorithms.

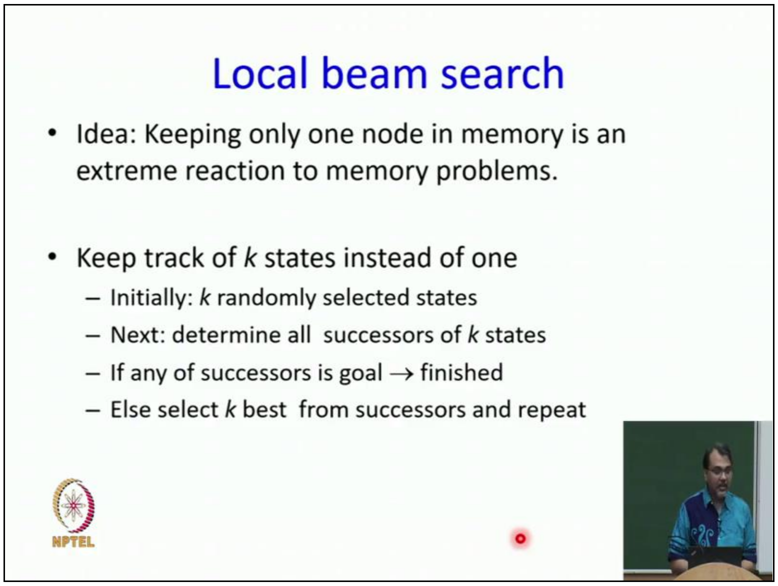
We are saying that at any point in time I have only one current node. Now go back to you know what we used to do in the previous systematic search algorithms exhaustive search algorithms we would do breadth-first search. And in breadth-first search we will have a full frontier of you know nodes and now we are saying that is taking too much memory so I will take one and that is an extremely action to the memory problem.

So can you think of an alternative algorithm which relaxes this is a assumption that I only have one current node. Simple question what is your name hash, hash says let us start with few random nodes not just one random no no this is not that deep sort of obvious but then if I let us I start with K nodes let us say I am simply doing greedy hill climbing. I start with K nodes I look at each of their neighbors and I pick the best.

Now what should I do how many best should I pick? K, we started with K let me pick K best can you guess which algorithm this is coming closer to; somebody is running ahead of me, we will come back to that but an algorithm that we have studied so far, randomly restarts is it coming closer to the random restarts by any stretch what is the random restart part here. Well you have a good point again I am going to come back to that but let us first give this algorithm a name.

So, go back to the algorithms that we studied where did we say that we will take at most K beam search exactly seen beam search we said lots of successes but I have limited memories I will give the K of them. So, this algorithm is called local beam search.

**(Refer Slide Time: 03:04)**



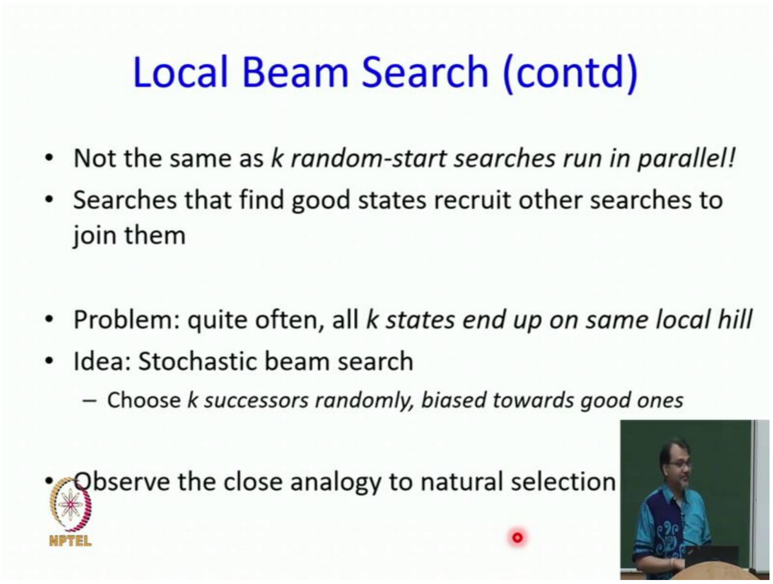

And it says keeping only one node in the memory is an extremely action to the memory problems let us keep K big deal. And initially I have K randomly selected states randomly start think about it next determine all the successes of K States as many. If any of the successes is good you are done of course if otherwise select the K best successors from this set of successors and that is your new beam. So, instead of just carrying one current node I have K event nodes at any point in time I have a beam of size K.

No, is it, so that is now supersedes this question is it equal to doing K random restart searches in parallel? Yes or no, how many of you think it is it is equivalent to doing K random restart searches. How many of you think it is not equal to doing K random restart searches person in the red what is the name what, why? It is possible that 2 or 3 of the next states comes from the same successor list of the same initial node.

So, this is not because what is happened going to happen it is possible that multiple next nodes are from the same hill. Now is that a good thing or a bad thing? It is a bad thing we started with a diverse set and then very soon we are moving towards 1 or 2 Hills, so our diversity is lost so how can we fix it? You can keep a limit on how many states you can keep from a certain hill that is a good idea what is the name? Yash says this another idea, so he says do not select K best select K by 2 best and the rest you choose bad options with some probability what is the name Ishant .

So, Ishant says select some of the best ones and for the others to some random probability jump and so this kind of an algorithm would be called a stochastic local beam search.

**(Refer Slide Time: 05:50)**



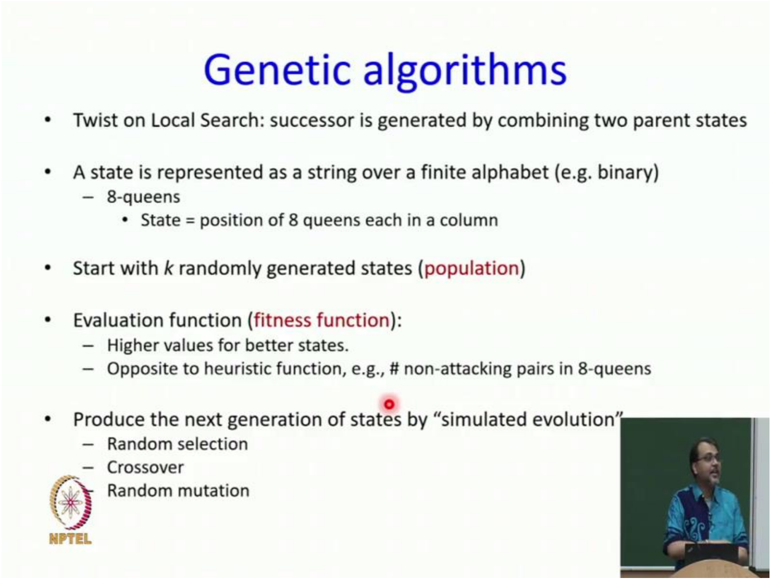

It basically says choose K successes randomly biased towards g ood ones and if you want we can also make sure that some of the top few months are also in it as you say everybody with me not too terribly complicated some it is sort of obvious. As you start building up an idea you start to get into these places. So, you start with local beam search you say instead of one I will keep K I will take all the neighbors take the best K but now all the best K are coming very close to in one hill I do not want to do that.

So, let me just randomly sample the successors with hire successors with a higher probability and worse successors with a lower probability. Now observe the natural analogy to analogy to the natural selection process that happens in the world. I have lots of insects and then the better ones live and the worse ones die. And then they have their children and the better ones way when the worst ones die this is sort of natural selection.

But this is like Ameba natural selection, why is this like Ameba natural selection and not like human natural selection sorry if we take successes I mean but the parent does not live anymore. Sure that is a good point but there is a better answer to this now I agree with you this you have a point what is your name? Poorva, so yes I agree but there is another thing I was looking for so how many parents does a child have?

In local beam search see I have many successes I am strong enough to create children I do not need anybody else to create a child uni sexual reproduction. Asexual reproduction as they say so. This is the idea of local beam search that I will create children and the better children will live. But now that we have reached this far let us create an algorithm that is similar to humans and their style of the production and so that algorithm is called genetic algorithms.

**(Refer Slide Time: 08:15)**




And also called evolutionary algorithms which is what Ishan was also referring to earlier and what are they going to change by the way if you think about it if you had to create genetic algorithms what would you say, how will change stochastic local beam search. I have a beam of size K what would I do, I will take what? Almost very close he says what is your name? Abidha, Abidha says I will have 2 successors and having somehow make their combination that could work too. But you know I am looking for the child who has two parents.

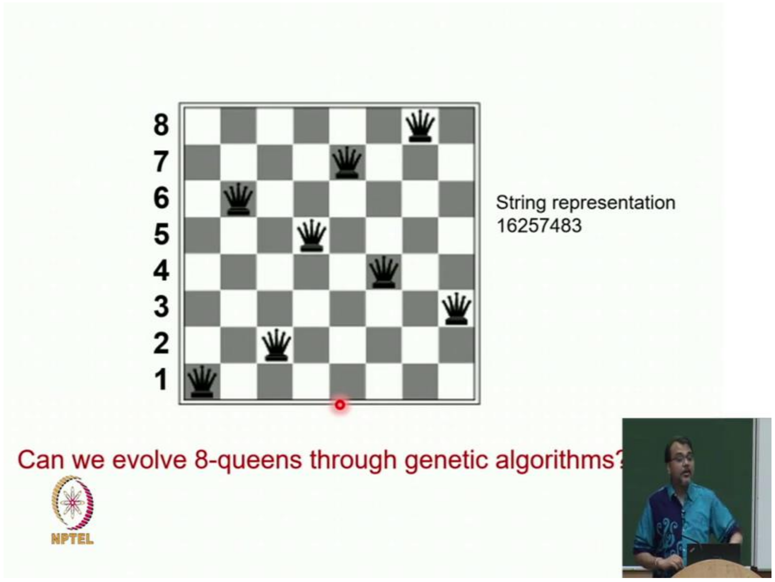
So Who am I parents the initial set of K, so how will I create my successors. I will somehow take 2 parents from this list 2 nodes from this list and somehow create a successor which has properties of both, so this is what happens I is like a father and you know lives like the mother that kind of thing, that is exactly what we want to do. And how is that achieved in humans what operation genes but what operation, I know you are engineering students your biology is really weak but somebody must know.

What operations between genes leads us to some property of the father and some property of the mother, pattern-matching you guys have to machine learning it come on get the genetics first. Look it is a very sick you know this is in 11th and 12th biology definitely you are taught but I think even in 9th and 10th you were taught. We have genes or we have you know all these chromosomes or whatever unites I do not even know my words right.

But some operation happens so that the final outcome is one of this and one of that sorry I cannot hear you but anyway the word I am looking for is crossing over, you have read this, crossing over bridge. So, that is exactly how we are going to create our genetic algorithm. So, we will create a state as a string so this is how the genetic algorithms work, It is nothing more than a twisting local search technically local beam search.

Where successor is generated by combining two parent states, a state is represented by a string over the finite alphabet.

**(Refer Slide Time: 11:12)**



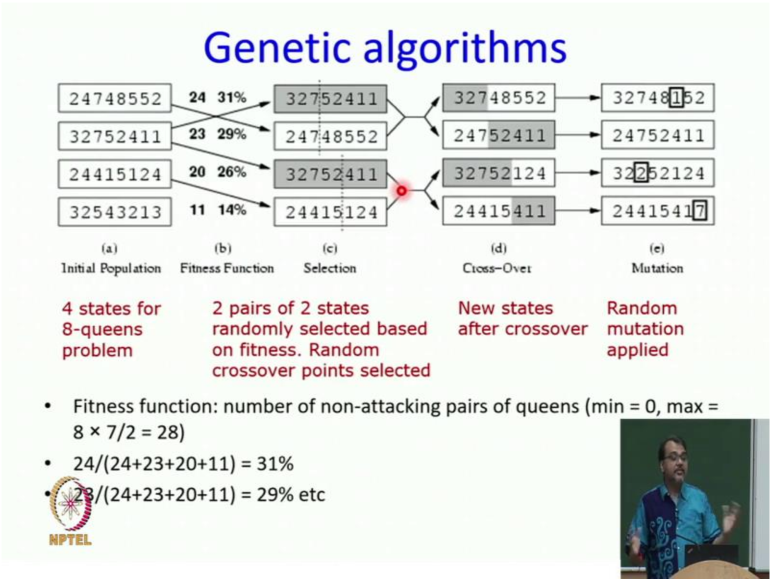

For example let us take this our n-queens problem or eight Queens problem we can say that we have numbered all the rows from 1 to 8 and all the columns from 1 to 8 and the six string representation would be for the first column where is the Queen for the second column where is the Queen for the third column where is the Queen like 1 6 2 5 7 and so on so forth. So, this string represents the state and now what we have to do is that we will start with K States because we have in local beam search.

So, I have K states like this each state has a string representation but in the world of genetic algorithms we are going to call it the population. These are all the humans in my current generation. Ok and we have associated a fitness function and the fitness function is that exactly the same fitness function or a slightly different then of what we are trying to optimize. So, for example in this case our fitness function is number of non attacking pairs in 8 Queens problem.

So the fitness function sort of does not change and now we are going to simulate evolution and so we are going to do 3 steps. We are going to do the random selection which is saying that actually fit parents survive the less fit parents die. And then they are going to marry them or they do not have to marry but you know they will produce children and they will and they will produce children by two steps.

One step is crossing over and one step is mutation random mutation which is also what happens in our gene pool randomly some genes change. And then we will create a next new population and the process will continue.

**(Refer Slide Time: 13:01)**




Let us understand this with an example, so let us say we have these 4 states as our population for each of those 4 states I can compute the number of non attacking pairs of Queens so this this has 24 this has 23 this has 20 and this has 11. So, which state is the least fit? The last state. Which state is the most fit? The first state very good, so now what we do I take all these numbers the fitness numbers and divide by the sum and that gives me the percentage of fitness I have to convert it into a percentage because I need to have a natural selection process which will flip a coin with some probability.

So, basically what happens in the first state has 31% fitness the second state has 29% fitness the third state has 26% fitness and the fourth state has 14% fitness. Now I randomly sampled 2 parents let us say I randomly sampled the second parent and the first parent, so this is I copy the second parent here and the first parent here. Now these are the gene sequences that I have to do crossing over. Over now if you understand what is crossing over you have one sequence you of the other sequence they cross which means first part of first parent and second part of second parent they come together to create a new string.

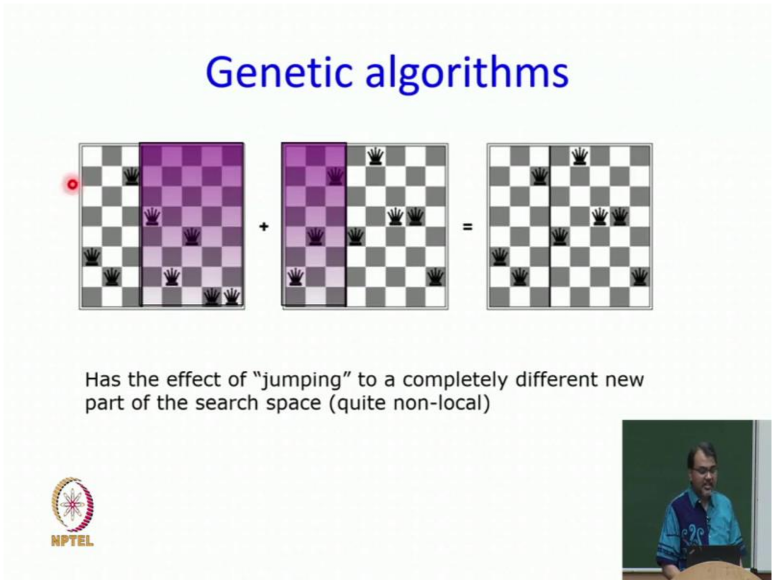
And that is exactly what we are going to do. So, for that we need to figure out a crossing over a point so let us say we randomly figure out that 3 is my crossing over point. So, now I will take 3 to 7 from the first and 4 5 5 2 from the second and create that as my new child. And I will take 2 4 7 from the second and 5 2 4 11 from the first and create that as my new child. And so I have now created these two new children 3 2 7 4 8 5 5 2 2 4 2 4 and if you think about it they are not close to either of the parents, does it make sense.

This as a string is a relatively different string from where what parents came it came from and then you can also simulate random notation so you can say over randomly I am going to flip this 5 to 1 it is a random gene change, so this 5 becomes 1, I do not do any mutation here maybe this 7 becomes 2 maybe I have 2 mutations somewhere. So, with a small probability I take a random thing and change it and this is like a simple successor.

What was happening in the simple successor I was changing only one Queen in that column that is this random mutation is not different from that it is exactly the same. But after this random mutation what I get I get 4 new children, so this basically I get a new population and I can repeat the process. So, where is the natural selection happening which step is the natural selection happening? First step when I am sampling a parent for reproduction that is when natural selection is happening.

I am less fit parent I am sort of giving less probability of being sampled where it is crossing over happening crossing over is happening on explicitly when I am combining the 2 strings where is mutation happening or where is the individual change happening that is happening at the last. Now intuitively what is happening let us think about it.

**(Refer Slide Time: 16:48)**




So, this was my first parent and this was my second parent and they individually were good but now what I did I took one part of this and the other part of this and created a chessboard out of it. So, you can think of crossing over as a completely non-local jump you know if go back to the vocabulary of local search. The vocabulary of local searches that I was making local changes in the local neighborhood that means if I am given one solution I was making only one change in the solution or very small change in the solution so that my new state is very similar to the original state hopefully with a better objective function.

Now I have sort of fog on that I am saying that I have two good states and I am crossing over in a way such that the jump that I make is a very huge jump in my local neighborhood. If I had to go from this state to this state it is going to happen in 5 moves and if I have to go from this state to this state it may have you know 3 moves or whatever. But this particular phenomenon is going to be really helpful because if it works for humans it should work for machines also.

Another way of saying it is that in typical local search we were only making small local changes and now we have allowed ourselves to make a very high jump in the search space that ought to be good for something. Moreover these are not random local jumps these are good local jumps because the original states were good. Now it is possible that we did crossing over at a point where the structure of the problem what lost and so the new state is really bad but does that hurt us? No in the next population you know in natural selection it is going to just die.

So it does not matter but it is possible that in this way we will suddenly hit on some very nice configuration which we which will develop further and lead me to a really good solution. You have to repeat so if I have a very good solution in my current population what do you want, so good question what is the name? Alkirath says that if I found a really good state in my previous iteration previous population then the cause of crossing over that will be lost is that a good thing? Of course not if you have found a really good state we should not lose it.

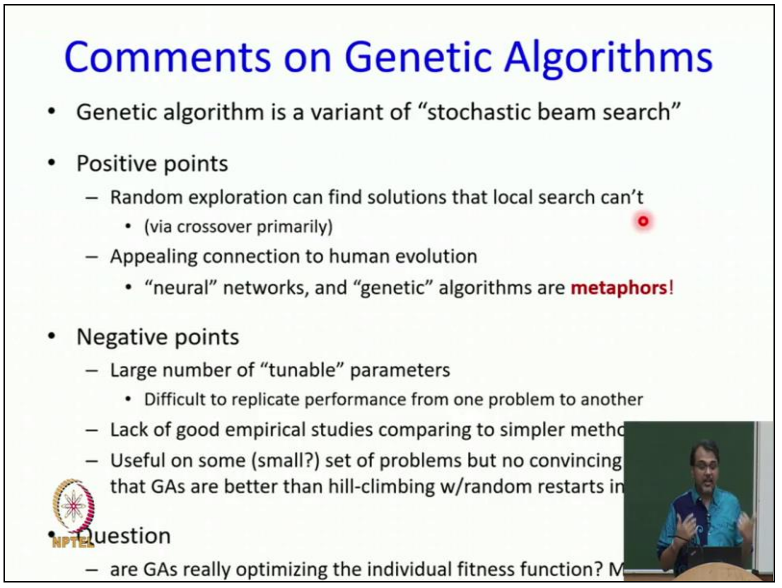
So how can we fix it yeah I just keep saying the best state forward or the best few states forward in the population. Those states are called champions in this vocabulary of genetic algorithm. So, just make sure that the champion moves forward and if a new champion comes in so it says. How do you say it is similar to humans? So, now the question is in which ways is it similar to humans and in which ways is it not similar to humans that is sort of the question you have to ask?

So if you think about how the child's actual sequence chromosome sequence or whatever it is comes out, the operations are very similar not allowing that. Now if we have children which are highly diseased then maybe very early in their life they die unfortunately so there is some little bit of natural selection there or they lead their life and you know not very effectively. So, that is where that happens and there is mutations in our bodies also so that is where they are similar.

Now how is it not similar well the champion never dies only in genetic algorithms that does not happen in real world what else the heuristic function that we are defining it is different from humans of course humans are not optimizing for the number of non attacking pairs of queens. Humans are optimizing for something else and the hope was and the belief was that humans optimizing also for fitness functions that you know fitter bodies or fitter people would do a better job of sending there; having their children in the next generation and that sort of gene pool will sort of takeover, over time.

No it is actually an interesting question whether that is the truth or not? So, let us talk about this now that we have understood genetic algorithms. Let us just quickly critique the genetic algorithm.

**(Refer Slide Time: 21:59)**




So, genetic algorithm is nothing more than a stochastic beam search it is just a variant of stochastic beam search which does its successor production in a certain way. The fundamental difference comes from you crossing over and not from anything else. Because the idea of random selection is already part of stochastic beam search the idea of mutation is already part or already part of the local moves in stochastic beam search.

The main idea that it brings to the table is the idea of crossing over. And that is the positive point that because of crossing over the random exploration can find solutions at local search cannot because of suddenly bringing together two solutions and jumping in a very far away space. And notice this whole idea of human evolution etc is metaphorical that makes sense to you. This is where the intuition comes but after that it is a specific algorithm. So, we will use the word neural networks but that is not how our brains are made we will use the word simulated annealing.

But that not exactly how liquids are cooled we will make the use of the word join together but actually people do not you know live their life in that particular fashion if they live like that it will be quite sad? Now what are the negative better points with respect to the algorithm itself it has a large number of tunable parameters. How many what size of the population should you start with how many parents do you create a new so you can do casting over 2 you can do crossing over with 3, you can do casting over with 4 we are no longer in the very nice world where a child has 2 parents a child can have many, many, many parents.

We are in the modern world and similarly you can you may figure out what is the mutation probability. So, there are lots of these knobs and a specific combination of the knobs can sometimes get you a very good solution but a slightly different knob may not give you away the equally good solution. So, the algorithms may be sensitive to these parameters. As earlier on there were a lot of good empirical studies but they were found to be successful on small set of problems.

But in general there was no convincing evidence that genetic algorithms are better than hill climbing with random restarts. Now I would also point out that this particular point of critique also comes from the politics in which the field of machine learning has evolved AIi has evolved in the last 15 20 30 years. Generating algorithms was a very important part of AI. I would say in the 80s and the 90s at some point people started seeing genetic algorithms researchers as people who are not formal enough people who are whose methods are difficult to reproduce.

Because of which there was a little bit of an ostracization or the genetic algorithms researchers sort of got ostracized from machine learning conferences. It is very interesting how politics of science also works with how the science moves forward. So, over time genetic algorithms folks created a different community completely and so they started publishing in this area called Gecko in this conference called Gecko and you would no longer find genetic algorithms based work in triple-A each guy neural science and so on so forth that dominantly.

Is it because genetic algorithms are not a good algorithm it is a difficult question for us to answer. In fact when the neural network revolution is happened in the last five years actually there is a little bit of the resurgence of evolutionary algorithms also. So, when ISO teach this algorithm five years ago I would often say that our genetic algorithms do not work. But I do not say that anymore eh because I know slightly better because I have seen that there is a historical reason why we were trained to say that genetic algorithms do not work better.

But also because in the latest times there are some successes that are coming out of the evolutionary computation community in the context of training user networks in the context of training reinforcement learning agents and so on. So, therefore I would say that jury is still out in fact if you read Pedro Domingues is the master algorithm which is a very famous book for machine learning person, everybody should read it.

He talks about 5 different tribes of machine learning he says all these tribes are trying to come up with the master algorithm, one algorithm that is going to solve all problems all machine learning problems and of course you know his ideas are you know do frequentist computations or do Bayesian computations or do you know neural network computations etcetera but one of the tribes that he talks about is evolutionary computation tribe, is the genetic algorithm tribe.

So at this point I would let you figure it out; you try it, it is one of another possible things that you can try in the world when you get come up of the new problem it is not my favorite I still go for that hill climbing with random restarts but there is some value the genetic algorithms off. The other thing I would say is that there is some research in the theory community which says that we as people are not optimizing for fitness and that is actually a very interesting question in and off of itself.

That in the world that we live in is it really happening that we are much fitter than what you know we were 100 years ago. In fact it is founded if some particular person has a certain combination of genes which is makes this person extremely fit there is a good chance that in the next generation that particular combination of genes would be lost and that confuses us because that is in conflict with the genetic algorithms thesis that if you keep doing this you will get fitter and fitter and fitter.

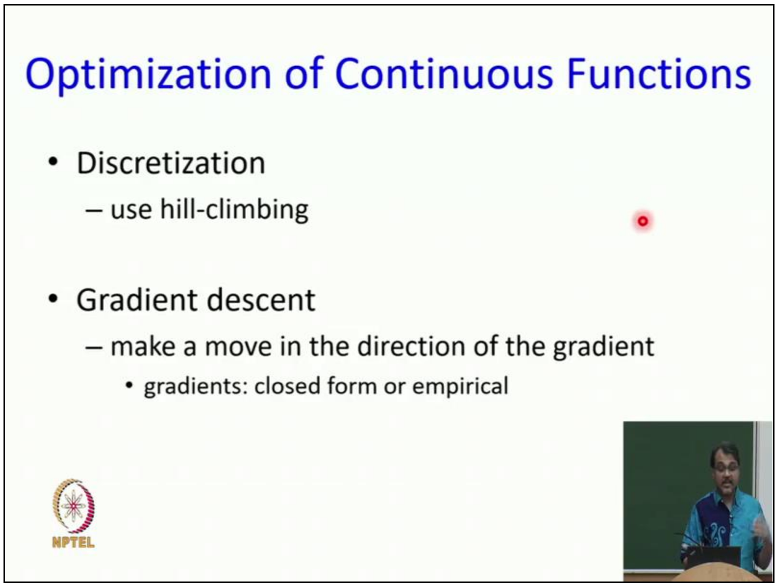
And so what these researchers have sort of found out by simulation and analysis is that we as communities as people or as organisms we are not optimizing for fitness. We are optimizing for this other phenomenon called mixed ability. And mixed ability say that if I find a random woman in this world to reproduce which there is a good chance that my child would be healthy I have the ability of mixed ability. I can mix with many, many people not just one person and that is something that we are optimizing because as people it is very hard to find the perfect match for us.

So the idea of that I can I still as want to spread my generations further and so I am saying that I can do it by mixing with different kinds of people I am not looking for a very perfect fit and so as an animal humans are optimizing for mixed ability and not fitness. And that brings back the question in the context of genetic algorithms where genetic algorithms must be really optimizing for fitness or something else we do not know that could be also a reason why they were not considered very successful in a lot of cases.

So I will keep my critique at this point I do not know all the answers but these are sort of the parts of the puzzle that I know with respect to genetic algorithms. Before they stop I will take 2 more minutes and say one more thing which is that local search is to me one of the most important algorithms in the field of AI. If I am given a really large problem any systematic search algorithm will fail more often than not because the problem size would be too high whereas a local search algorithm would be able to give me something a good answer more often than not.

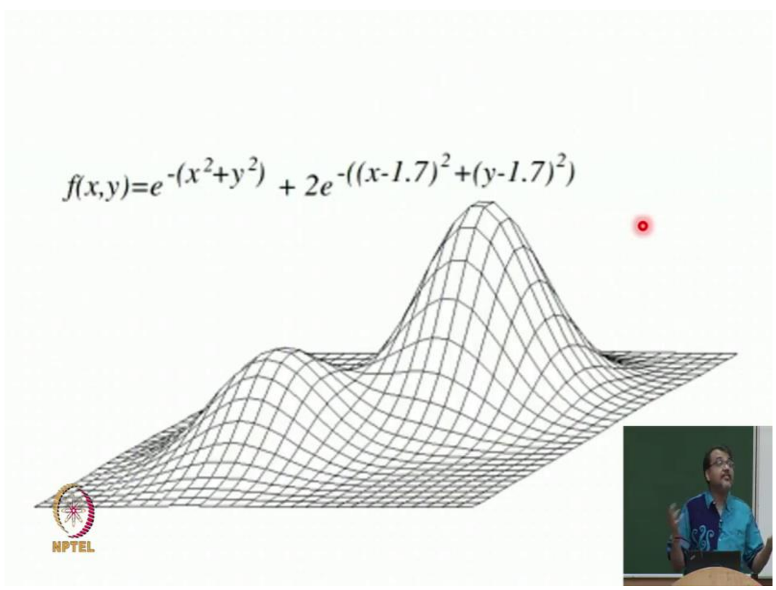
And in fact this thing becomes even more salient and prominent when they work in the world of continuous variables. So, in local search our variables were discrete and we were making local discrete moves.

**(Refer Slide Time: 29:54)**




But suppose we are in the world of continuous functions a we can always use discretization and hill-climbing but nobody does that what people end up using is gradient descent or gradient ascent. But that is nothing more than the local search version in the continuous space.

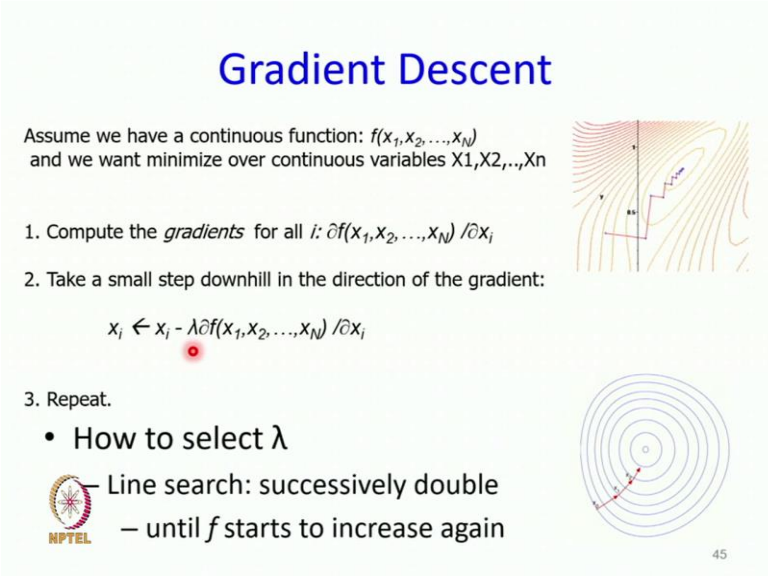
**(Refer Slide Time: 30:06)**




Because if you have a complicated function that you are optimizing think a neural network or any kind of machine learning algorithm then it will have some kind of a topology there will be some Hill that I am interested in reaching I am interested in teaching a certain X Y value and now if there are too many values and it is a very complicated function then how do I optimize it well I take the derivative with respect to X take the derivative with respect to Y and make moves in that direction.

And again the intuitions are similar if I start here and start making moves I might be stuck in local optimum. If I start here and start making moves I might reach the global optimum. So, all the intuitions that we have learned in local search get somehow applicable in the context of optimization although their form changes a little bit.

**(Refer Slide Time: 30:48)**




For example in gradient descent we first compute the gradients then we change each variable in the direction of the gradient. Now what is the notion of gradient the notion of gradient is various my closest neighbor in the greedy direction that is the idea of the gradient eventually. What is the idea of learning rate the learning rate is how much do I change not that is different because in discrete we were making discrete moves, so we did not need that the amount that we will change it.

Now we have this learning rate this is this lambda and it tells me how much do I move in the direction for gradient if I am doing ascent or direction opposite the gradient if I am doing the same. And there are many algorithms to figure out how to select lambda like one algorithm is you start with X you add lambda and then you add 2 lambda then you add 4 lambda and you keep doing it until your objective function is improving.

And at some point it starts reducing then you stop and you take the previous lambda and that is your lambda that you want to output this is called line search. So, the main point here is that intuitively at some meta point most popular algorithms in machine learning in AI are doing some kind of local search this is not always true A star is not local search A-star is still useful there are other algorithms which are useful.

But a lot of modern machine learning algorithms at least are local search in continuous space and that is called gradient descent or this. So, with that I will stop this completes our lecture on local search and in next class we will start talking about how to play games.