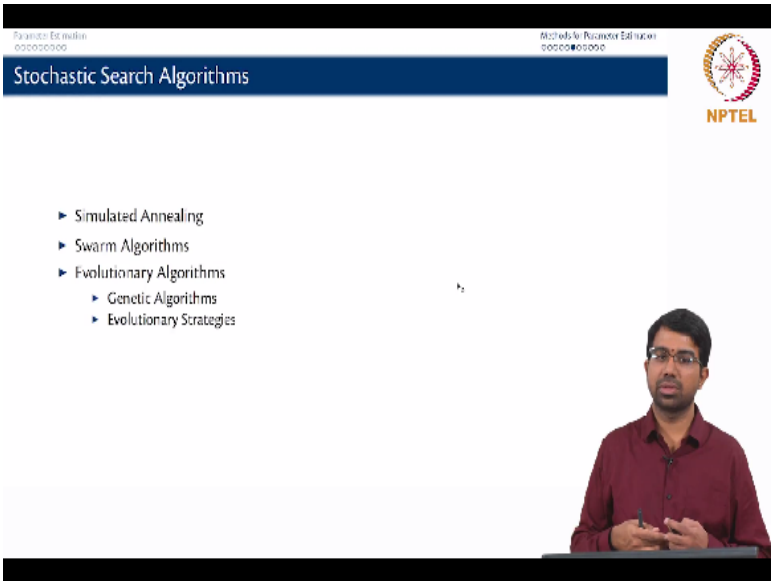
Computational Systems Biology
Karthik Raman
Department of Biotechnology
Indian Institute of Technology - Madras

Lecture – 44
Direct Search Methods

(Refer Slide Time: 00:11)

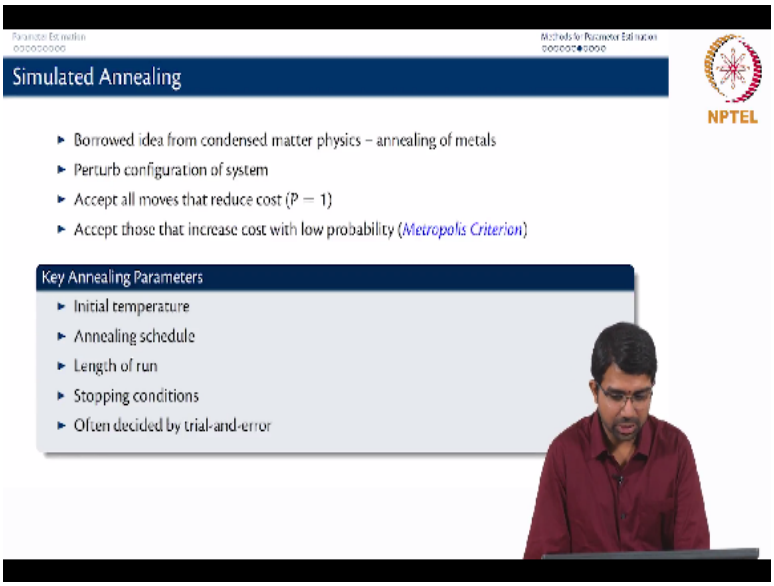
In this video, we will overview some of the direct search methods that are used for parameter estimation or for any optimization problem in general. We will look at stochastic search algorithms, what are stochastic search algorithms and 2 classic algorithms namely simulated annealing and genetic algorithms.

(Refer Slide Time: 00:29)



So the classic stochastic search algorithms, some of you may have heard of, are simulated annealing. There are particles swarm algorithms. Of course, a battery of evolutionary algorithms including genetic algorithms, evolutionary strategies, differential evolution and so on.

(Refer Slide Time: 00:43)



So let us look at simulated annealing which is one of the most, one of first methods in fact and a very popular method for non-linear optimization, complex optimization. But I must warn you that simulated annealing has not been found to be too useful in case of biological systems because it is, it is expensive and more than being expansive, the convergence rates have been quite poor, at least in the study that we did a few, a few years ago.

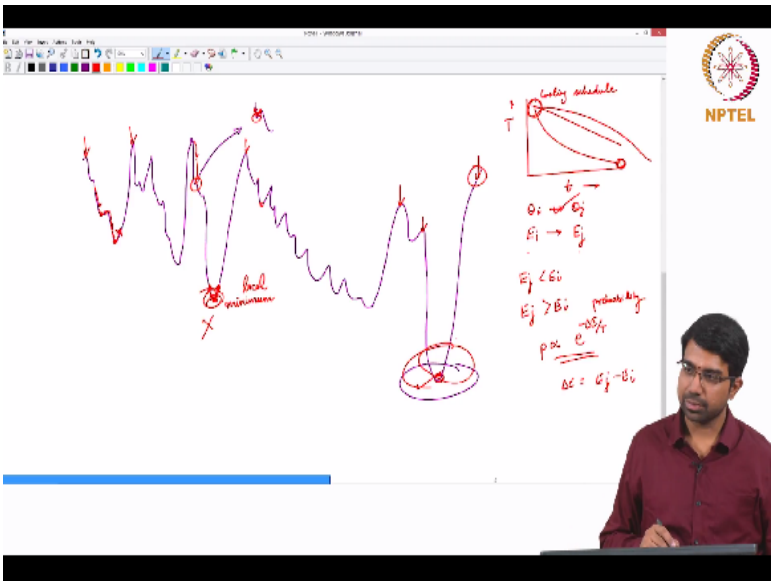
So simulated annealing essentially is inspired from condense matter physics. So you all may have heard of annealing. It is a common metallurgical process. So what is annealing involves. You essentially heat up a metal to very high temperature and slowly cool it to room temperature. So what it does is, it tries to remove the imperfections in the metal and give you nice metal at the end of the process.

So how does that relate to computations? So you need to look at analogy. So what is heating, what is temperature, what is cooling and so on. So the key idea in simulated annealing was the fact, so what happens to the molecules in the metal? They have a lot of kinetic energy, right. So they are really jumping around at the initial high temperature plus you start cooling the system, the molecules kind of settle down, right.

So the same idea is used here. What you do is, you start searching for an optimum by heating up the system, meaning you are very exploratory in the beginning. You really jump around the parameter space and then you slowly cool the system and you stop jumping around and you really commit in some sense to exploring a particular direction and array but 1 minimum. Again, it could be a local minimum but the idea is because you have done a much better search, you are likely to get at least a better local minimum rather than even if not the global minimum.

So how does it work? The key idea is, you perturb the configuration of a system and you accept all moves that reduce the cost. This is what is your classic hill climbing, no difference in hill climbing, accept all moves that reduce the cost but the key innovation was, this was known as the metropolis criterion, you accept a small fraction of the cost, of the moves, that increased cost, with low probability.

(Refer Slide Time: 03:19)



Meaning you accept everything that reduces cost which will bring you here but then with a small fraction, with a small probability, you might be able to get out of this point and then will again continue to descend and may be reach a better minimum but again with the small probability, you might be able to go, go out of this but what you usually do is, you have something called a cooling schedule and may look like this or look like this or look like this.

What is the rate at which you cool down the system? So what you end, let us say move from Ei to Ej, right. So you move from essentially theta I to theta j, the cost of which is Ei in the first case, Ej in the second case. If Ej < Ei, accept this move for sure, no question, right. That is like a classic hill climber. If Ej>Ei, you accept it with some small probability. This probability is generally proportional to e to the… So this is again very inspired from Boltzmann distribution, condensed matter physics.

"Professor - student conversation starts" (()) (04:57) So delta, yes, -delta i, so delta E is basically Ej-Ei, right. So E to the -delta E/… (()) (05:10) **"Professor - student conversation ends."** So what happens, when t is very low, you will reject a lot of moves that go in the wrong direction. When t is very high, you will accept a lot of moves that go in the wrong direction. So you are very exploratory in the beginning.

So you kind of jump all around the place and then after a while you probably settle down and

hopefully reach this point. **"Professor - student conversation starts"** So what is T here, T is what? Temperature. (()) (05:50) yes, some notion of temperature. It could well be a number of iterations, right. I mean, no, so your T is basically the number of iterations. This is some parameter that tells you, we need a cooling schedule.

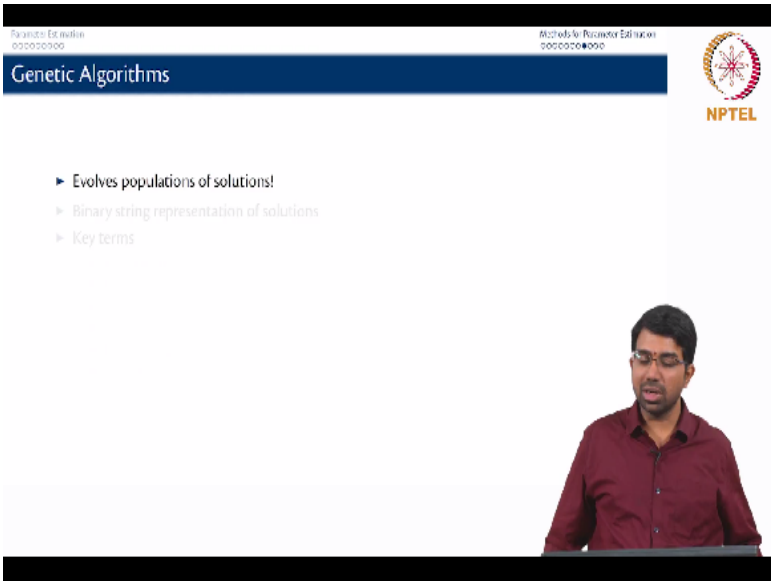
It could be linear in which case it is basically like number of iterations or the inverse of it because with increasing number of iterations, you want a lower temperature. **"Professor - student conversation ends."** So what are the annealing parameters and what do we mean by annealing parameters? They are the hyperparameters for your, your estimation algorithm. Your estimation algorithm here is simulated annealing which has certain parameters you have to tune. So any algorithm first of all has to have a stopping condition.

When do you say stop, stop exploring, stop jumping around, right. So when T comes to 0, may be but, but you may also want to stop when you have, your E becomes low enough, the error, the objective function. So suppose you are minimizing something, the fit of your data to your experiments and so on, so may be if that number becomes very low, right, you are within 1% error of the dataset that you have, then no problem. Stop the annealing.

So initial temperature, annealing schedule, the length of the run, stopping conditions and so on and often these are decided by trial and error, right because it is not easy to estimate these for different datasets. So may be for a particular kind of dataset, you know one kind of cooling schedule will work. For another kind of dataset, another kind of cooling schedule will work. Why would this be the case?

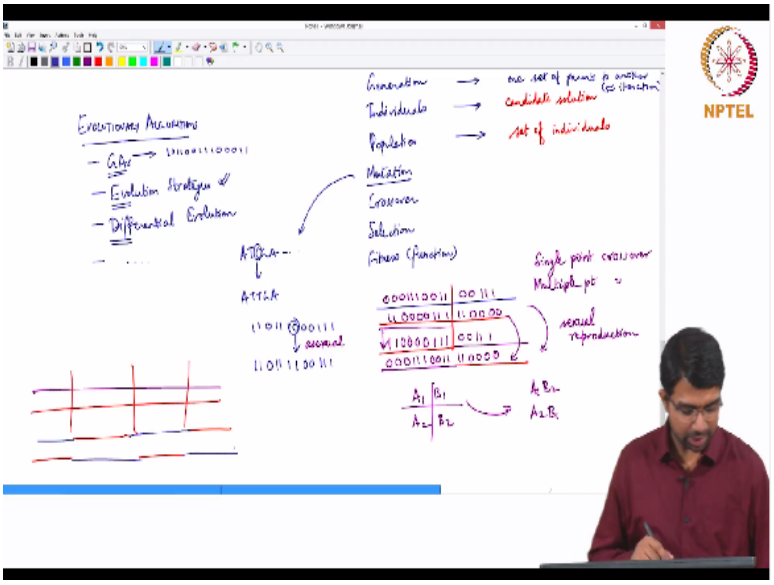
"Professor - student conversation starts" Because there are lot of (()) (07:32) depending on the data. Depending upon the data and the model, right. **"Professor - student conversation ends."** Because the landscape depends upon the data and the model. What is your land? What do you mean by landscape? It is E of eta. The cost function of theta, right and e of theta in turn depends on x which in turn depends upon the model, your predictions and the predictions in turn depend on the model and of course the parameter values. This can get quite tricky.

(Refer Slide Time: 08:10)



So then look at genetic algorithms which is again a very popular, you know, direct search algorithm which is extensively used in parameter estimation. You see most biology papers will report parameter estimation using a genetic algorithm and in genetic algorithm you have a few flavours. So genetic algorithms in, they are all broadly part of evolutionary algorithms.

(Refer Slide Time: 08:40)



And inside of these you have genetic algorithms, you have evolution, strategies, very common used in biology. Then you have differential evolution and several others, right. You can have an evolvable hardware and multiple other things. The most important for us are genetic algorithms, evolution strategies and differential evolution. So for this first we need to understand various concepts of evolution.

The first important thing about evolution is, it happens only in populations. So you need a population and every population is essentially made up of individuals and how does evolution occur? It happens via mutation, crossover and such. So it involves mutation, crossover and then there is selection that is important and selection happens based on fitness. In fact, you call it here a fitness function.

So all these are essentially biological terms which have new connotations in the context of evolutionary algorithms. So what is an individual? **"Professor - student conversation starts"** It… It is essentially a candidate solution. **"Professor - student conversation ends."** Population is groups of, okay, set is better. Set of individuals, okay. There is also another important term here, generation. You all know what generation is, right?

So parents belong to one generation, children belong to the next generation and so on. Similarly here. You have one set of individuals belong to one generation. From those individuals, you generate other individuals, right. So there is some reproduction that has to happen and then you will get the next generation and so on, right. The population is…

"Professor - student conversation starts" It is the iteration (()) (11:43) Yes, you can say it is like iteration. It is the initial condition. Initial condition is initial population. Generation is number of generations will be a hyperparameter. But how can it have a group of initial solution, initial condition, I will have. Random, random X0, right. **"Professor - student conversation ends."**

So all of these, all direct search algorithms actually need a theta 0 to start with. How do you pick that theta 0 is also important? Usually random, right so and it has to, in fact in genetic algorithms one, we just add to the random initial population. Start with the random initial population, apply evolution crossover and so on and put a fitness, selection pressure in the right direction. Let us see how we go about these?

So what is mutation? Mutation is essentially in biology; we have a sequence like this. If this

becomes this, that is a mutation, right. One base that is changing. This will cause, this can subsequently cause some change in the protein, may be a change in function, gene exploration and so on, right. Whereas here in this case, for GAs, everything is represented as bit vectors. If you have a bit vector of this sort, so bit vector basically is made of bits 0s and 1s.

You want to mutate it. So what you do is, you randomly pick a spot to mutate and you flip the bit. Here this 1 bit is flipped, right and in crossover, you essentially have 2 stretches of genes, right and you chose a point to crossover and then you get 2 genes that are modified versions of the parents. So this you can think of as asexual reproduction and this is essentially sexual reproduction. Basically getting information from 2 parents here; whereas here, it is just coming from 1 parent.

So depending upon the case, this might be like a global search and a local search if you are looking at the real space, okay. So what is crossover mean? So you have 000111001100111 then 110000111110000. So this will now become, no it takes the whole thing. So this will now become 00011100111100, sorry, that should have come down, so that, this will now become 000111001100, sorry 110000.

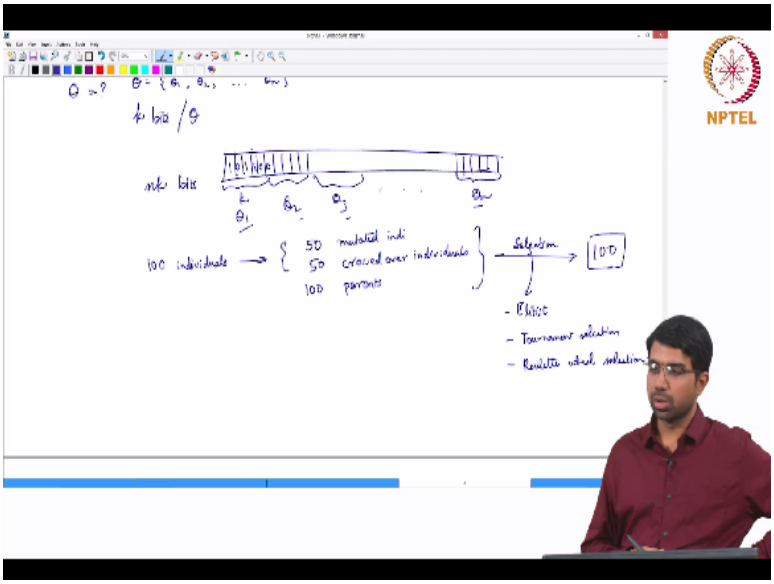
And this becomes 00111110111000011, right so this fragment comes directly here and this fragment goes here and so on. So essentially if you have an A1 B1 and A2 B2, you have a crossover which gives you A1 B2 A2 B1 if these are segments of the genome or here in this case, segments of your bit strings, bit vectors.

"Professor - student conversation starts" (()) (16:42) You randomly pick, you can have. So this is actually what is known as a single point crossover. You can have a multi, multiple point crossover as well, right. So basically let us say you have a gene that looks like this and another gene that looks like this, you have multiple points, so basically this goes here, this goes here, this goes here, this goes here.

(()) (17:26) Exactly. So we still not gotten into what is the exact representation. So what we said is this is a way to represent solution to some optimization problem. In our case, it is going to be

the parameter estimation problem for which we will have a specific representation, we just come to that. (()) (17:49) each of the bits represents parameters? **"Professor - student conversation ends."** So let us, let us look at it.

(Refer Slide Time: 17:56)



So what is your bit vector because what is it that you want to estimate, you want to find out what is theta? And let us say theta=…, right. So you want and let us, let us say we use some k bits per theta. You will end up having nk bits, right. So you will have a long bit string. The first k will be theta 1. The next k will be theta 2. The next k will be theta 3, theta n. Now you flip many bits here, right.

So you are going to change theta 1, theta 2, theta 3, theta n. Many of them will change or you take 2 such vectors and cross them over, right. So some theta 1 value will go to theta 4 or you know you can get, you will get some mixed up versions of the 2 parameter vectors, right. So there are various ways in which you can do this. So you have a lot of freedom and variety in how do you pick these operations and how do you go about each of these operations.

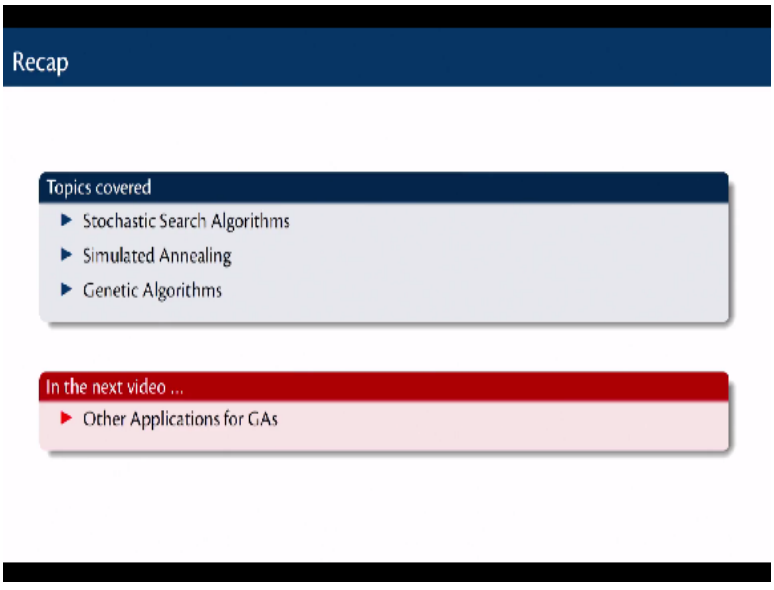
"Professor - student conversation starts" (()) (19:33) Yes, so, so let us just recap the situation. **"Professor - student conversation ends."** So in, we have, we start with 1 generation with 1 population, right. At the end of 1 generation, you do a bunch of mutations, crossovers and so on and generate multiple children. One thing that you do in most evolutionary algorithms is that you

keep the population size fixed. So let us say I started with 100 individuals. I mutate them, do whatever I want, right and I might come with inter, you know, in the middle.

I might have 100 mutated individuals and I might cross them over and so on and produce another 100 and may be I also have the 100 parents, right. Now again select 100. So you throw away 200 solutions, right or you might just create 50 mutations, 50 crossovers and throw away 100. There are various recipes you can come up. Each one you can, you can try any number of variants that you want, right. You finally have to examine, test these in terms of how good the convergence is and so on.

"Professor - student conversation starts" But throw them (()) (21:30). So you, how do you do the selection? You just have the maximum (()) (21:40) thrown them away. So this is, this is tricky. **"Professor - student conversation ends."** So how do you actually come up with the best way to select the next population, right. So there are various selection methods that are available, right. You can have what is known as elitist selection, can have what is known as tournament selection, or roulette wheel selection.

(Refer Slide Time: 22:23)



So in this video, I hope you had a good introduction to direct search in stochastic search algorithms in general and in particular simulated annealing which is a very interesting way, which is one of the oldest methods proposed to jump out of local minima in optimization

problems as well as genetic algorithms which brings in a new flavour by adapting concepts from evolution to computation. In the next video, we will look at some other applications for GAs such as in scheduling and so on.