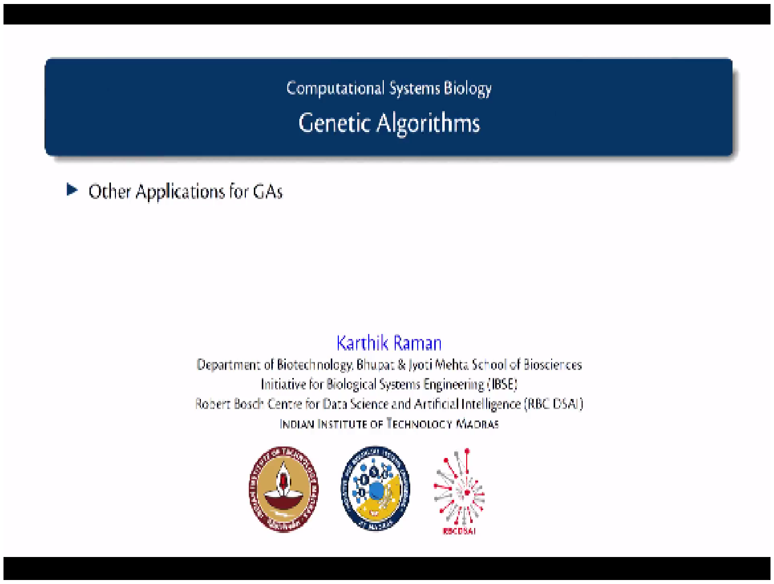
Computational Systems Biology
Karthik Raman
Department of Biotechnology
Indian Institute of Technology - Madras

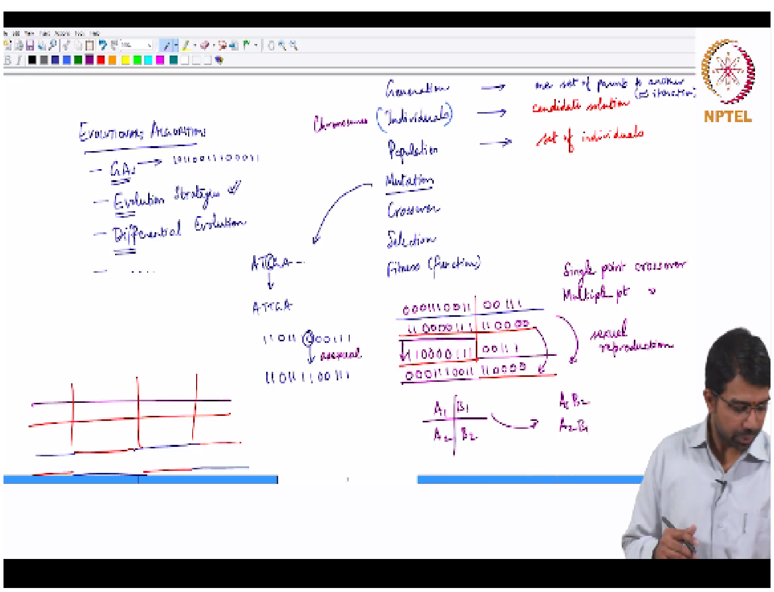
Lecture – 45
Genetic Algorithms

(Refer Slide Time: 00:11)



In this video, we will digest a little and look at other applications for GAs such as in scheduling problems.

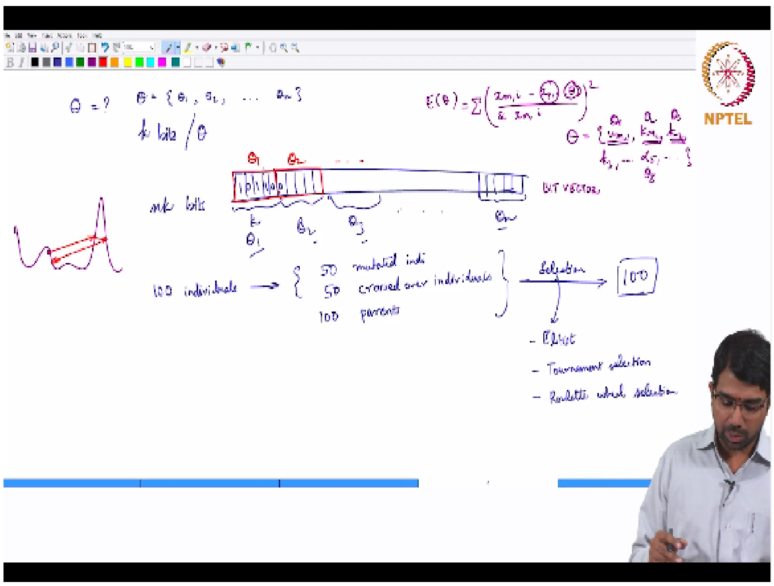
(Refer Slide Time: 00:18)



Welcome back, we will look at genetic algorithms. We will continue from where we left off in the previous class. Let us look at some examples for how we can use genetic algorithms to estimate parameters and once again, you know, clarify all the fundamental concepts of how genetic algorithms work.

We looked at bunch of concepts yesterday, essentially those of what is a generation? What are individuals? Or in fact, they are also called chromosomes and the concept of what is a population? Then mutation crossover selection and so on. And how do we apply this to a practical problem, right?

(Refer Slide Time: 00:580)



What is a practical problem that we have on hand? We need to estimate? We have this sort of a cos function, e of theta is some, something of this sort, right, where each of these x, xpi's are actually computed by integration, solving a bunch of differential equations, then capture your mass action kinetics or (()) (01:42) in kinetics and so on. You need to find out this theta. What is theta?

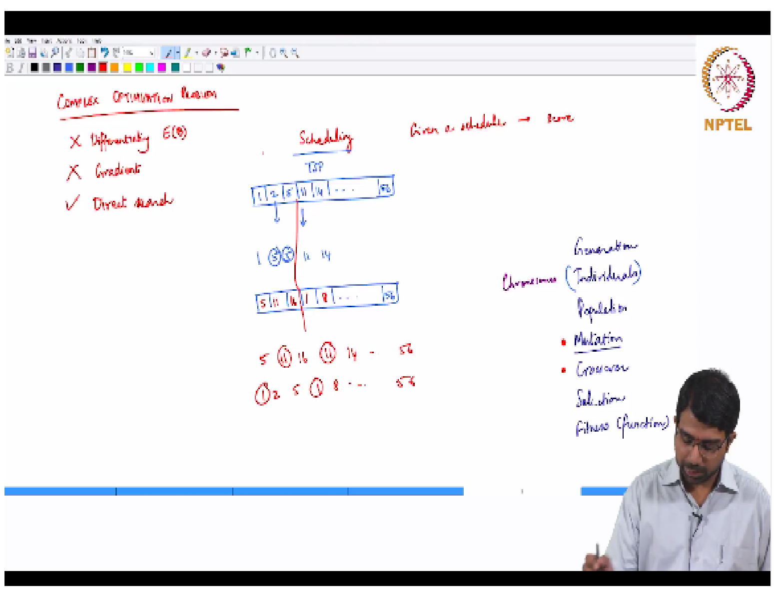
It is nothing but Vm1, km1, then could be some k2, k3, alpha 5, whatever. All the parameters that describe your mathematical model, your biological system. It could be your particular binding affinity, it could be your rate constant, it could be an enzyme turnover number, whatever with maximum rate of reaction. So all of these are your parameters. Now you can think of sticking them into a vector of this sort so wherein, so this is like a bit vector where let us say you are contiguously putting in theta 1, theta 2, theta 3.

So this is basically theta 1, this is theta 2, this is theta 3, this could be theta rho sum 8 and so on, right. How many your parameters are there in the systems and let us say you have n parameters in the system and you just put them all next to each other. Now this is the point, so this is the equivalent of the point that is being explored, right. From here, you make a mutation, you probably jump here.

You make a crossover, you probably jump here. You make another mutation, you probably jump here and so on, right. This is what it practically means. Of course, this is a point in n dimensional space and here we are talking about single dimensions but the idea is the same, right. So you basically embed all your parameters, this is theta 1, theta 2 and so on. There are other ways to represent.

So in A genetic algorithm, if you just take a step back to see how genetic algorithms work, the biggest challenge is that of representation. So let us pause and take a look at what are the other domains where genetic algorithms are used.

(Refer Slide Time: 03:59)



The genetic algorithms can be used in any domain where you have a somewhat difficult complex optimization problem where basically you have given up on, you are given up on differentiating the cos function or you have given up on gradient and you are relying on direct search alone, right. So one classic example of this sort is scheduling. So in fact, there are universities that use some genetic algorithms to schedule the final exam timetable, right because different students have different courses that they are taking.

And how do you make a final example timetable, the final exam timetable that does not you know have too many exams for the same person on the same day or even overlapping exams, right. So suppose you are taking course 1 and course 2, you cannot have both courses having the final exams at the same time, right. So how do you resolve these clashes?

So what you could do is, given a schedule, I can compute a score, how good that schedule is or how bad that schedule is, right. I might be able to look at the number of overlaps or you know whatever, things like that, right. But what would be even mean by computing a differential of this or by trying to compute the gradient of this, right?

“Professor - student conversation starts” (()) (05:59) It is not even if I take x variable, right. So it becomes very difficult. **“Professor - student conversation ends.”** So given a schedule, you can compute a score. So how do you do this mutation and crossover. You need some more take, one schedule, convert it to another schedule, or you want to take 2 schedules and mix them up to form a third related schedule, right.

This will be mutation and crossover. So your regular words, your regular concepts will have, will end up having different meanings depending upon the problem in question. So what were the problems that we were looking at? What were the parameters that we were looking at? You were looking at... So these were the concepts that we were looking at earlier. So what is a generation? I should probably call it evolution algorithm concepts, right. These concepts are shared across all flavours of evolution algorithms. We did look at at least a few, right.

So there is genetic algorithms, there are evolutions strategies, there is genetic programming, so on and so forth. There are many variants but in all of these cases, you need a recipe for each of these. What is a generation, what is an individual or a chromosome, what is a population, what is a mutation, how do you do crossovers, how do you do selection, how do you compute fitness? You need a method for each of these things.

So let us look at scheduling. So in fact another example I usually give is how do you schedule the IPL, right? So you may want to schedule the IPLs in such a way that you do not have too many free matchless days for a particular theme, right. You normally find in the IPL schedule that the same team might go, might be playing on 2 consecutive days but then they may not have a match for like 4-5 days.

So I have trying to give an example of IPL, right. So how do you schedule? So similarly you pick (()) (08:32) schedule and you have a schedule, you can compute some sort of a score but thing is, how do you plan your representation? So let us say this is actually the same issue that one comes across when you solve the travelling sales person problem, right. So the travelling sales person problem or any, like IPL scheduling problem is something like...

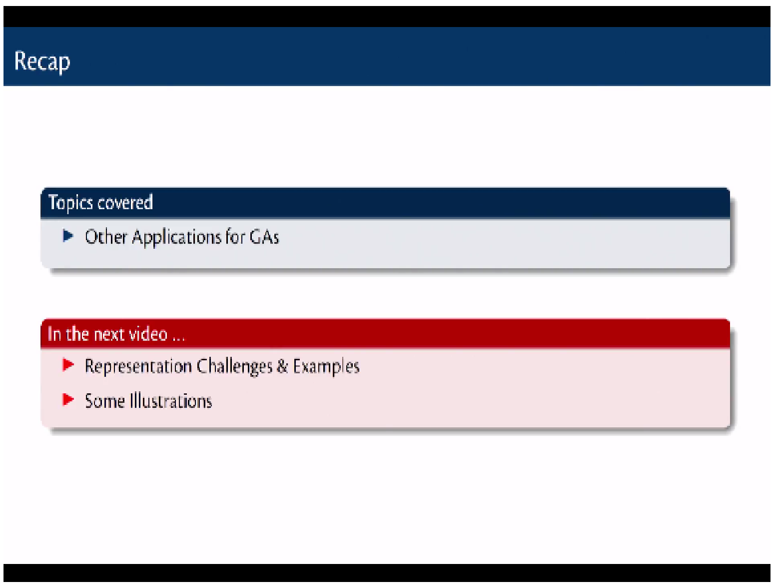
So let us say there are 56 games. I want to order the games, right. So I may be know which game has, there has to be, the game and the city is not going to change, right. You need to have a particular game in a particular city or you need to have a particular sales person visit a particular house and make a particular delivery. How would you order it in such a way that something is economized?

This is classic scheduling or travelling sales person problem. So now if I were to do a mutation here, I just cannot change here 2 to 5. I will have two 5s which is not acceptable, right or I just cannot take a copy of this vector and crossover with this, right. So let us say I crossover at this point, right. I would get a schedule 1 2 5. Okay, these are identical. So let us, let us actually take a slightly different schedule.

Let us say 5 11 16 1 8 something like that. Let 56 remains as is. So now the 2 schedules I will get are 5 11 16 11 14 56 and 1 2 5 1 8 56. Again this is not what you would consider a legal schedule, a valid schedule. So you need to essentially come up with some other equivalent operation for mutation and for crossover, right. What would be the equivalent operation, you will have to figure out. So, so may be you just want to swap. Not a pick 2 matches and swap them. You wanted to pick 2 cities or pick 2 deliveries play points and swap them, right. So there are different things that you can do.

“Professor - student conversation starts” But again on that depend on the matches next day and the previous day. Why? Is it like you cannot get 2 match gap. No, no, so then we will score that schedule. So you create, all you have to worry about is creating valid schedules. You know, making, having sensible representations. **“Professor - student conversation ends.”**

(Refer Slide Time: 11:28)



I hope you got an overview of what are all the possible applications for GAs. We are through this video and in the next video, we will discuss a very important aspect of using GAs to solve optimization on parameter estimation problems which involves how do you go about getting a useful representation that is enable for running GAs on it and we will also look at some examples and illustration.