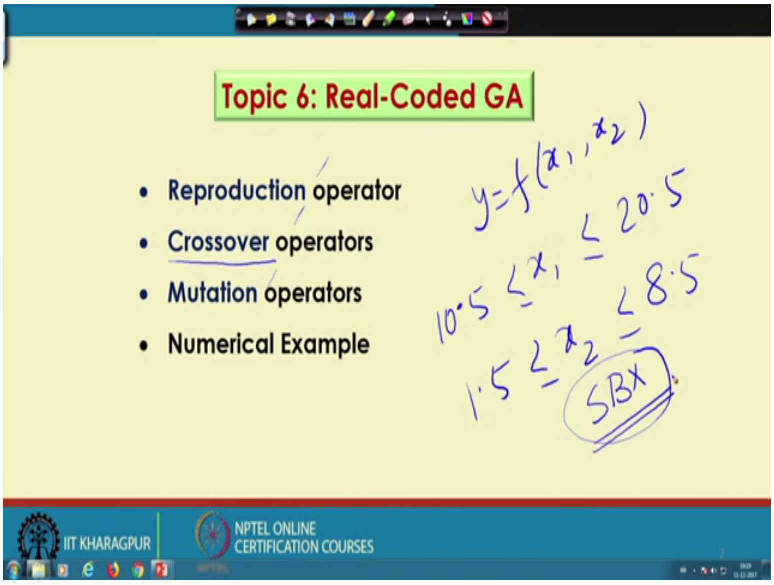
Traditional and Non-Traditional Optimization Tools
Prof. D. K. Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 45
Summary – III

Now, topic 6 it is on real coded GA. Now suppose is that in the optimization problem the variables are real in nature so, there having some fraction.

(Refer Slide Time: 00:29)



Topic 6: Real-Coded GA

- Reproduction operator
- Crossover operators
- Mutation operators
- Numerical Example

$y = f(x_1, x_2)$

$10.5 \leq x_1 \leq 20.5$

$1.5 \leq x_2 \leq 8.5$

$\boxed{SBX}$

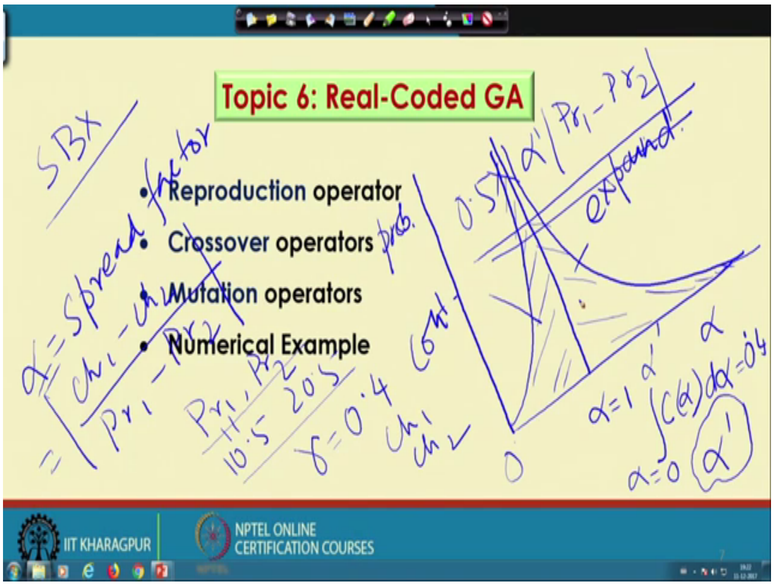
For example, say, say I have got a function of say 2 variable, y is a function of 2 variable say x 1 and x 2. Say x 1 is lying between say10.5 to say 20.5, and x 2 is lying between. So, 1.5 to say 8.5 so, both the variables are real. And if I want to solve the optimization problem involving this type of variable is better to go for the real coded GA.

Now, here actually what it do is, we try to take the help of the same set of operators like you are the reproduction crossover and mutation. Now generally we use the reproduction operator like, the tournament selection as a reproduction operator for this real coded GA. Now let us try to concentrate on this particular the crossover operator. Now if you see the literature, we generally use different types of crossover operators in real coded GA. Like for example, if you see some of our earlier application, there we use the concept of some set of linear crossover, then comes here the blend crossover, then the simulated binary crossover. So, these are actually the popular crossover operators, but there are a few other crossover operators also.

Now, out of these actually the simulated binary crossover could gain much popularity. Now the working principle of this simulated binary crossover; that is, SBX I have already discussed in much more details. Now to summarise the principle of this particular the SBX, let me mention that here the probability of creating the children solution from the parents. That is expressed with the help of some function, some polynomial set of function and we try to define something for the contracting zone of this particular crossover, and expanding zone of this particular the crossover.

Now, let me let me try to explain little bit once again.

(Refer Slide Time: 02:55)



Topic 6: Real-Coded GA

- Reproduction operator
- Crossover operators
- Mutation operators
- Numerical Example

Now supposing that we have got the probability distribution something like this; now before that we defined a particular. So, I am discussing in short, the principle of SBX. So, we define a particular parameter that is alpha that is called the spread factor. Now here the spread factor is nothing but the ratio of the difference between the children solution to the parent solution the mod value of that.

So, this is nothing but the spread factor. So, if I plot spread factor along this, and the probability distribution along this particular the probability distribution function along this. So, what you can do is; so, corresponding to a particular value of alpha say alpha equals to 1 alpha equals to 1 a particular value. So, we try to define some distribution some we take the help of some polynomial. Now supposing that so, from here to here, here alpha equals to 0 to 1; so this is called actually the contracting zone, and alpha equals to 1 to infinity that is called the expanding zone.

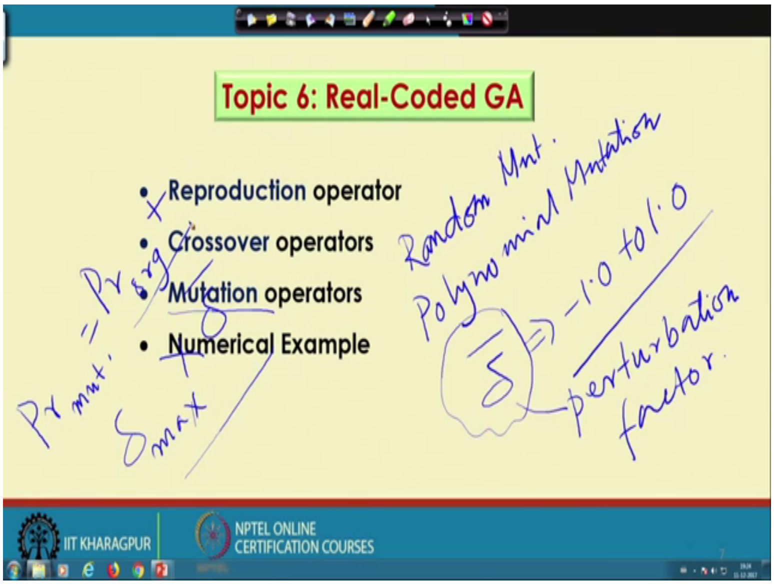
So, this is actually the contracting zone and this is actually the expanding zone. So, the area under this particular the curve, the total area will be equal to 1, and the area under this particular the contracting zone is 0.5 and in the expanding zone once again 0.5. Now here actually what you do is, supposing that I have get 2 parents parent 1 and parent 2. The I have got some numerical values say one could be10.5 another could be 20.5 something like this. And using these 2 parents I will have to find out what is child 1 and child 2.

So, what you do is we try to generate some random number lying between 0 and 1. And that particular random number is going to indicate the area under this particular curve. Now supposing that it is in the contracting zone, supposing that the random number r is found to be say equals to 0.4. So, if it is 0.4 so, what you do is we try to find out the integration alpha equals to 0 to alpha prime. Then this contracting zone function is c alpha d alpha so, this particular integration is equal to we put is equals to 0.4. And we try to solve this particular your alpha prime. And once I have got this particular alpha prime, that mean particular value for this spread factor. Now using that I can find out your the child 1 and child 2.

Now, to determine these child 1 and child 2; so what you do is, we consider the average of the parent properties, and then we try to find out the difference of the parents p r 1 minus p r 2 the mod value of that multiplied by alpha prime multiplied by 0.5. So, this 50 percent we add to a particular child and the 50 percent we subtract from another child to get another child. So, this is the way actually we can implement you are the SBX.

Now, using this SBX we can find out the children solution, and once I have got the children solution, now we can go for the mutation operator. Now if you say the literature we use different types of mutation operator in real coded GA. Now out of all such operators the most popular one one is the random mutation.

(Refer Slide Time: 06:40)



Topic 6: Real-Coded GA

- Reproduction operator
- Crossover operators
- Mutation operators
- Numerical Example

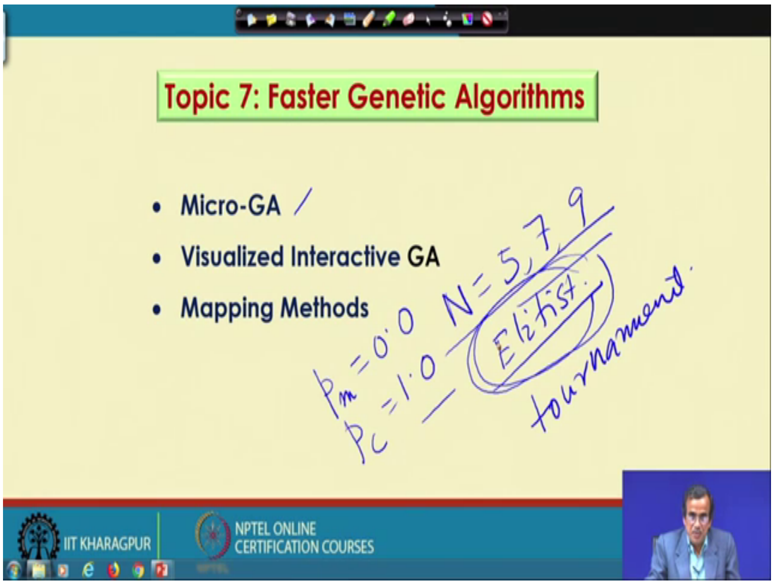
Random Mnt.
Polynomial Mutation
$\bar{\delta} \Rightarrow -1.0$ to 1.0
perturbation factor.
$P_{r\,mnt.} = P_{r\,org} + \bar{\delta}\,\delta_{max}$
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

We have got the concept of the random mutation, then we have got the concept of polynomial mutation. Now here in polynomial mutation, once again what we do is we take the help of one mathematical expression of this polynomial just to indicate what should be the pop perturbation factor that is delta bar.

Now, this perturbation factor is going to vary from minus 1 to plus 1, and we try to find out a numerical value lying in this particular range. And once you have got this perturbation factor, perturbation factor is nothing but the dissimilative factor, perturbation is nothing but dissimilarity. So, perturbation factor so, depending on the value of this particular perturbation factor. So, we find out the your the parent mutated is nothing but the parent original or original plus your delta max that is a maximum value of this perturbation, and your perturbation factor is delta bar. So, using this particular formula so, we can find out what should be the mutated solution.

Now, while discussing this real coded GA in detail, actually we solve the numerical example just to show, the working principle of this particular the crossover and mutation. So, just like the binary coded GA so, this particular real coded GA can also find out the globally optimal solution through a large number of iteration using the operators like your crossover mutation and so on.

(Refer Slide Time: 08:29)



Topic 7: Faster Genetic Algorithms

- Micro-GA /
- Visualized Interactive GA
- Mapping Methods

$N = 5, 7, 9$

Elitist

tournament

$p_m = 0.0$

$p_c = 1.0$

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

And using this actually we can find out this optimal solution.

Now, actually I am just going to concentrate on topic 7 that is faster genetic algorithm. Now we have already mentioned that the GA is computationally very expensive and it is very time consuming; that means, if you want carry out optimization online within a fraction of second. So, this type of ordinary genetic algorithm is not going to help you. And that is why will have to go for some faster genetic algorithm, like your this micro GA.

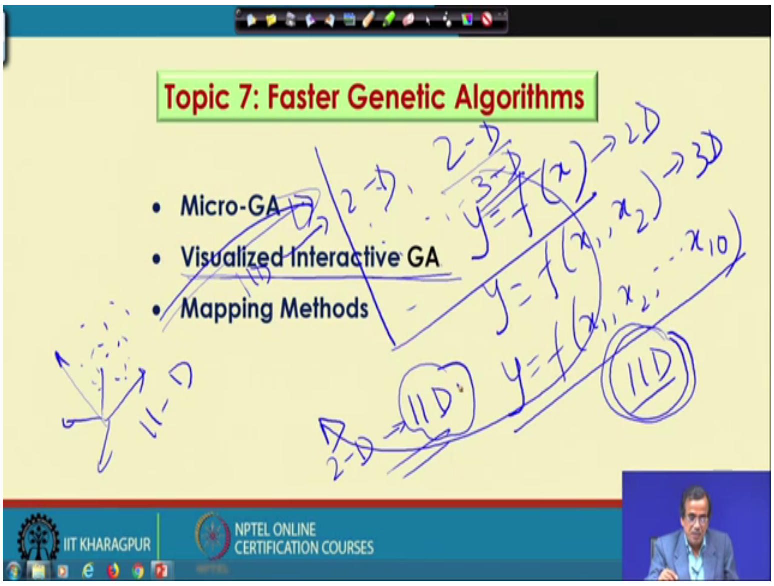
now this micro GA is actually a typical genetic algorithm or we use a small population size, and generally this population size is kept equal to say 5 7 9 something like this, some odd value. And here we take the concept of this elitism. So, use the concept of elitism, and this is a small population genetic algorithm. By using this particular elitism so, we can keep the already found solution in the next generation, and we use the small population just to make it faster, and just to inject the diversity in this particular population we take the help of the tournament selection.

So, actually we do the tournament selection, and initial solution is selected at random, and at the beginning of the next generation. So, one solution using the principle of elitist we take from the previous generation, and the remaining 4 of the next generation there is a second generation those are generated at random using the random number generator. And this process will go on and go on, and generally in this type of GA we do not use any such mutation. So, p m is kept equal to 0 and the crossover probability is generally kept equal to 1. And using this there is a possibility, that it will be able to find out the optimal solution within a fraction of second, but these particular optimal solution may not be the globally optimal solution it could be near optimal solution also.

So, this is actually the principle of this particular the micro GA. Now next let us try to concentrate little bit on this visualized interactive GA. Now here the purpose is actually how to make it faster just to help to find out what should be the most efficient the search direction. Now to determine the most efficient search direction actually what will have to do is, will have to investigate the nature of the objective function.

Now, as I have already mention that for any optimization there are 2 important things one is the step length another is the search direction. So, here actually what we are going to do, using the information of the surface of objective function we are going to decide what should be the most efficient the search direction at a particular the iteration. Now what we do is now we try to investigate the nature of this particular the objective function.

(Refer Slide Time: 12:10)



Topic 7: Faster Genetic Algorithms

- Micro-GA
- Visualized Interactive GA
- Mapping Methods

$y = f(x) \rightarrow 2D$
$y = f(x_1, x_2) \rightarrow 3D$
$y = f(x_1, x_2, \ldots x_{10})$
11D
2-D
3-D
11-D
2-D → 11D
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Now supposing that I have got a function of only say one variable. So, the objective function is in 2 dimension so, we can visualize.

If it is a function of say 2 variable x 1 x 2, it is in 3 dimension that also we can visualize, but more than 3 dimension we cannot visualize. Supposing that I have got a function of 3 variable; that means, the objective function is in 4 dimension. Or if it is a function of say 10 variable, say x 1 x 2 up to say x 10 so, this is in 11 dimension.

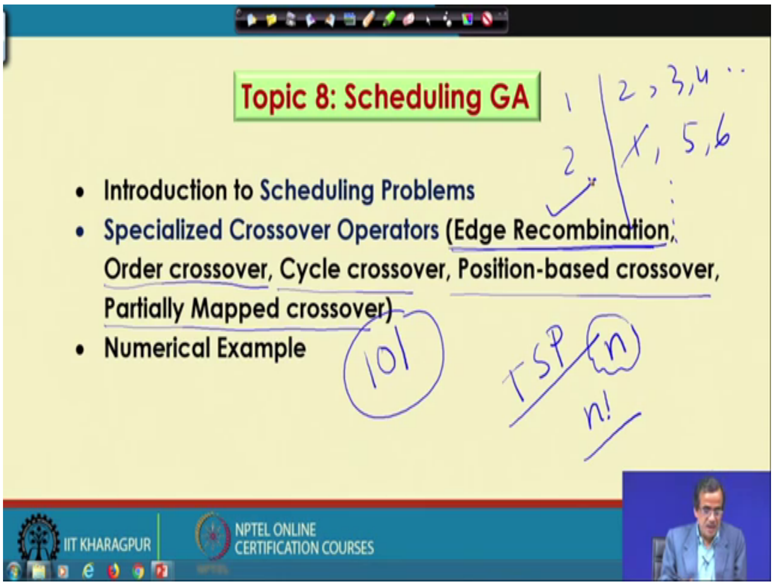
So, 11 dimension we cannot visualize, that is why the data from this particular the hard dimension. So, those are to be mapped. I have got a large number of points in say 11 dimension. So, these points are to be mapped in 2 d. So, just to just for the purpose of your the visualization. So, what we do is, we try to take the help of some mapping methods. So, using those mapping methods, this higher dimensional data are approximately map to the lower dimension for the purpose of visualization so that we can give some information to the GA that you try to concentrate your search only in this particular the region. So, here the working principle of visualized interactive GA is very simple. So, what we do is, supposing that GA is working on a higher dimensional space, it is trying to find out the optimal solution in say 11 dimension.

Now, so, 11 dimension we cannot visualize. So, what we do is; so, this 11-dimensional data or map to the lower dimension say either 2 d or say 3 d for the purpose of visualization. Now if I get this 11-dimensional data in 2 d and 3 d. So, I can see the topology of this particular your the objective function in 2 d or the 3 d. Now using the information, that in which direction the value of the objective function is going to increase, if I can collect that particular information so that information I am just going to a give to the GA by doing some setup rebuts mapping.

So, here what we do is, from 11 dimension to do 2 dimension we go. Here we do the rebuts mapping that is from 2 dimension to the 11 dimension, because the GA is working in 11 dimension. So, we the user is going to give information to the GA, that you concentrate in this particular region, and this 2-dimensional data will be converted to the 11 dimension, because GA is working on this particular the 11 dimension. So, by doing this there is a possibility that the user is going to help the GA little bit; and that means, the user is going to supply some intelligence to this particular the genetic algorithm; that is the principle of the visualized interactive genetic algorithm. And this visualized interactive genetic algorithm is supposed to be faster compared to your ordinary genetic algorithm, because some problem information to decide the most appropriate search direction is given to the GA in the form of intelligence by the user.

So, there is a possibility that visualized interactive GA will be faster. And we have seen that that this particular GA is found to be faster, but; however, it is performance depends on your objective function or and this is found be function depended.

(Refer Slide Time: 15:53)



Topic 8: Scheduling GA

- Introduction to Scheduling Problems
- Specialized Crossover Operators (Edge Recombination, Order crossover, Cycle crossover, Position-based crossover, Partially Mapped crossover)
- Numerical Example

The next topic is actually the scheduling problem. Now here scheduling problem is actually a special type of optimization problem, which is much more difficult compared to the ordinary optimization problem like say, design of gearbox design of spring and so on.

Now, here in scheduling problem actually will have to consider the not only the position, but will have to consider the order and adjacency. Now let me take a very simple example. Supposing that there are small n number of cities, and these small n number of cities are to be are troubled by a particular person only once by covering the minimum distance or by taking the minimum by consuming the minimum travelling cost. And he should be able to touch each of this particular city once. So, this is a typical TSP problem travelling sales person problem, and to solve this particular problem, what will have to do is we are going to consider n cities. And at the same time will have to consider the relative the position and the adjacency of city 2 with respect to the city one.

Now, here actually what he do is so, if there are n such cities, and will have to find out the sequence of this particular the travelling. There are n factorial possible sequences, and it is a huge number, supposing that I have got 10 cities. So, if there are 10 cities we have got 10 factorial possible sequences. And GA will have to find out GA will have to search all the possibilities, and by considering all the possibilities GA will have to find out this is the globally optimal solution.

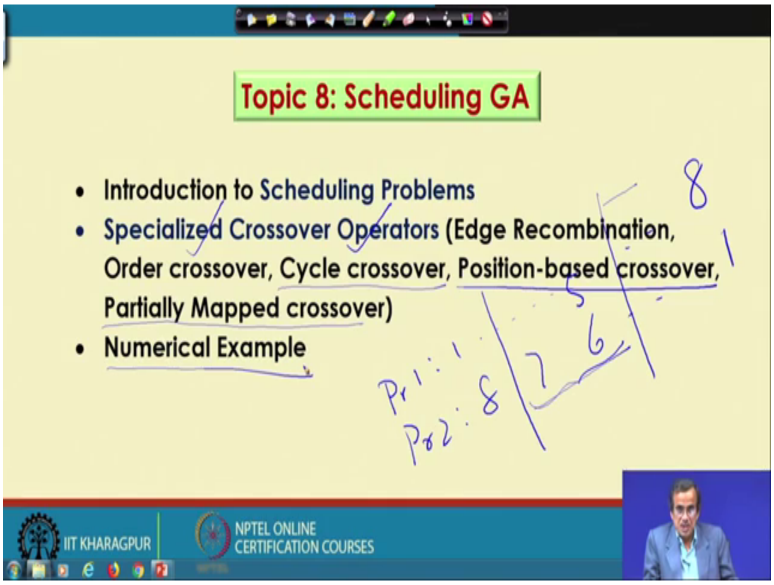
Now, here actually the GA is going to face a more difficult problem. Moreover, if you use the ordinary crossover operator; like, say single point crossover, 2-point crossover, multi point crossover or say uniform crossover there is a possibility that will be getting some in feasible solution; that means, like if we use a ordinary crossover operator there is a chance that will be getting some solution which is infeasible in terms of this particular the connectivity of the cities.

So, those things are to be overcome. And that is why actually we took the help of the special type of operator like your edge recombination, then comes the order crossover, then comes the cycle crossover, and then position based crossover, then partially mapped crossover. Now these crossover operators I have discussed in details with the help of some numerical example. But let me spend little bit of time on the principle just to summarise. Now in edge recombination actually what we do is, we just try to maintain one connectivity table or the edge table.

And starting from the 2 parents so, I will have to find out to valid children. So, we start with the starting city of parent one, just to get the starting city of child 1. And whatever cities we have taken so, we maintain one connectivity, next city one is connected to say 2 3 whole something like this. 2 is connected to 1 5 6 something like this. So, this type of connectivity table we have; the moment we select a particular city. So, from the right-hand side of this particular connectivity table we remove that particular the city. And these processes will go on and go on and ultimately, I will be getting one valid sequence for the children.

So, this is the way actually we implement the edge recombination. Now then comes your order crossover. Now here actually what you do is, we try to take the help of we try to take the help of say, say 2 parents, parent 1 and parent 2.

(Refer Slide Time: 20:29)



Topic 8: Scheduling GA

- Introduction to Scheduling Problems
- Specialized Crossover Operators (Edge Recombination, Order crossover, Cycle crossover, Position-based crossover, Partially Mapped crossover)
- Numerical Example

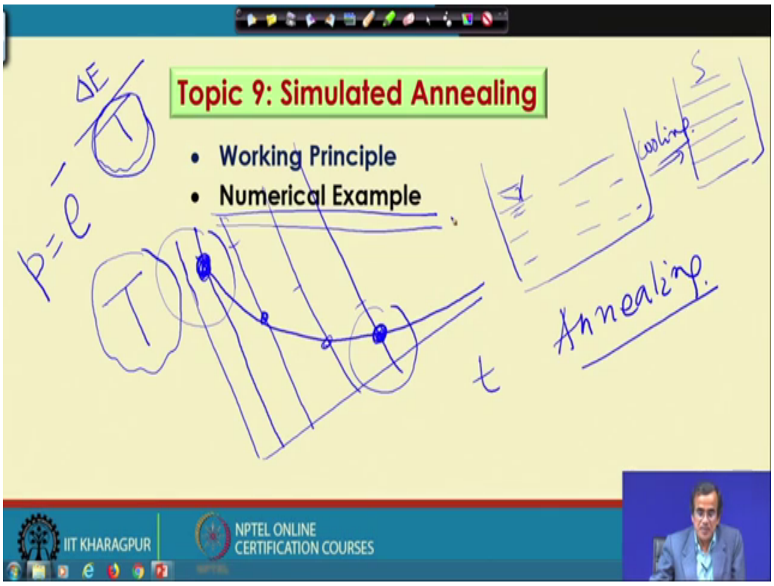
Supposing that the parent 1 is having a few cities like 1 2 3 4 5 6 say 8, and parent 2 is something like say 8 7 6. Let us last is a 1.

So, what we do is this order crossover, we try to select 2 crossover sides at random. And to get the children solution so, something we copy directly from the parent, and the rest of the things we take from the other parent. So, this method we follow, and this have already discuss. So, in much more details with the help of some example; that is the principal of order crossover. Then come to cycle crossover, we start with one starting city of the cycle selected at random. And so, we start with that particular starting city, and we will just continue this particular process and this particular cycle is going to be complete. And once it is completed, the other remaining position we take from the other parents.

So, this is in short, the principle of the cycle crossover, then position based crossover, we select a few position, on this parents at random, and with the help of the selected position we carry out the crossover, just to get the children solution. Then comes the partially mapped crossover, once again we take like 2 crossover sides at random, and to get the children solution a few cities we copy from say parent 2 the rest of the things will have to take from parent one following a particular the sequence or a particular the method. That I have already discussed in much more details, and we have solve some numerical examples also.

Now, the message which I want to convey, that if you are going to solve the scheduling problem, scheduling optimization problem, like optimal sequence of TSP or the optimal sequence of machining, supposing that I have got n job in machines. N job a machine, or say 2 job n machines that type of thing. So, this type of problem can be tackled with the help of this type of scheduling GA.

(Refer Slide Time: 23:01)



Topic 9: Simulated Annealing

- Working Principle
- Numerical Example

$p = e^{-\frac{\Delta E}{T}}$
T
t
cooling
Annealing.
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, this is actually the principal in short of this scheduling GA. The next topic actually we try to concentrate on some other very popular on conventional or non-traditional optimization tool. Now here the concept of simulated annealing actually what he do here is, a we try to model the molten metal the solidification process of a molten metal in the artificial way just to solve the minimization problem. Now if I consider the molten metal.

If you see the energy state of the molten metal so, this energy state of the molten metal if I consider. So, here this is at the highest energy state. So, this molten metal has to be a convert converted to the solidified one. So, this is a solid, and here we use some cooling process.

Now if I go for the first cooling so, for the still there is a possibility. So, we will be getting some polycrystalline state, and if I go for slow cooling so, will be getting the single crystal and if I compare the energy level of the polycrystalline state, and energy level of the now single crystal. The single crystal is at lower energy level, and that is why we go for the slow cooling. Now the method of slow cooling is nothing but the annealing. So, annealing is a process of slow cooling of molten metal.

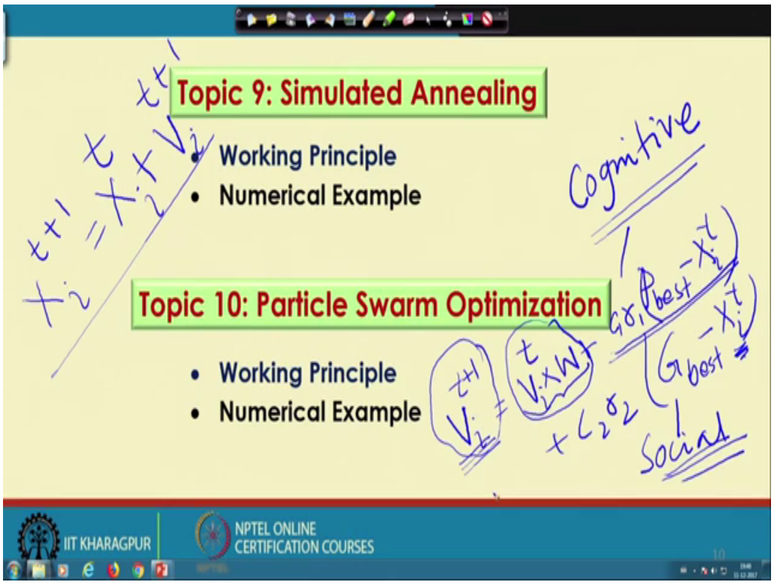
So, this particular actually the method has been copied in the artificial way in the simulated annealing just to solve the energy minimization problem. Now if I see the iteration of this particular simulated annealing. So, this is say number of say iterations, and let me considered so, this is actually so, the temperature of the molten metal; so initially so, the temperature of the molten metal. So, the temperature of the molten metal could be here. The next iteration what you do is we try to we try to reduce the temperature to the 50 percent so, it will be here

So, it will be here, the next iteration the your the temperature will be reduced to 50 percent, next iteration temperature will further reduced to 50 percent and so on. And if I just feat one curve, this shows the variation of temperature. And in fact, actually what you do is, we take the help of the Boltzmann probability that that concept, the Boltzmann probability and if the temperature is high according to the Boltzmann probability, the probability of selecting the bad solution will be more. Now here actually what you do is, the probability p is nothing but e raise to the power minus delta e divided by t, and k that Boltzmann constant; that is considered as equal to say 1.

Now, here actually what you do is as temperature decreases. So, as temperature decreases so, what will happen? 1 divided by this so, as temperature decreases. So, what will happen is the as temperature increases, let us see what happened, as temperature increases so, this here the probability of selecting the bad solution will be more, and has the temperature decreases the probability of selecting the bad solution will be less. And using this particular principle, that initially we can accommodate a few bad solution, but the probability of accommodating the bad solution will be less with the as the iteration proceeds.

So, this particular principal will follow just to minimize this particular the energy. And on principle it can solve the minimization problem. So, this is the way actually we we try to model the annealing process in an artificial way. And it can solve that particular the minimization problem, and we solve some numerical example just to explain actually just to explain the working principle of this particular the simulated annealing.

(Refer Slide Time: 27:38)



Topic 9: Simulated Annealing

- Working Principle
- Numerical Example

Topic 10: Particle Swarm Optimization

- Working Principle
- Numerical Example

$x_i^{t+1} = x_i^t + V_i^{t+1}$
$V_i^{t+1} = V_i^t \times W + c_1 r_1 (P_{best} - x_i^t) + c_2 r_2 (G_{best} - x_i^t)$
Cognitive
Social
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, after that we discuss the principle of particle swarm optimization. Now this particle swarm optimization is another very popular tool for the optimization. Now here just like a genetic algorithm, we start with a swarm of particles; that is a population of solution. And here we try to generate the position and the velocity using some principle, what we start with some initial position and initial velocity, and then we try to update that particular velocity, and we try to update this particular the position.

Now, what we do here is, we try to update the velocity as I told, supposing that the velocity at the t plus 1th iteration. So, we try to find out the velocity at t-th iteration multiplied by the weight that is the w. Next, we consider some cognitive component that is nothing but c 1 r 1 multiplied by that is called the p best, p best this I am going to explain minus. So, value of the variable say x, x i t plus c 2 r 2 the globally best g best multiplied minus x i t.

So, this is the way actually we are going to update this particular the velocity. Now here so, this particular term is going to give initial momentum to this particular the particle. And this part term is nothing but c 1 r 1 p best minus x i t is going to give the cognitive component. This is called the cognitive component, cognitive component and this is going to give the social component. So, here the meaning of this cognitive component, the social component and the initial you're the momentum. So, using these 3 things so, the particle is going to update this particular the velocity. Now by this cognitive component we mean, that each particle will compare with it is previous best just to updates it is own condition own situation. And at the same time, the particle will have to compare with a globally best, and this globally best is going to attract that particular solution toward better mind. That is nothing but the social component.

So, using these 3 components of initial momentum, cognitive component, and the social component; so the velocity of the particle will be updated. So, might be this is v I t plus 1 ok, and once I have got this particular updated velocity, now I can find out that x i t plus 1; that the position of the ith particle a t plus 1th iteration is nothing but the x i t plus this particular v i t plus 1. Multiplied by t t is nothing but one here, and that indicates the number of iteration.

So, this is the way actually we can update the position. So, the initial position velocity are generated at random, for the whole swarm for all the particles, and these particles are going to update their velocity and position. And through a large number of iterations so, this particular algorithm is going to hit the globally optimal solution. Now if I compare, the particle swarm optimization with genetic algorithm. So, this particle swarm optimization has got a few advantages. For example, say, we have already mention that genetic algorithm is a very powerful tool for global optimization, but it is local search capabilities poor. On the other hand, this particle swarm optimization is a very efficient tool for both global as well as the local search.

So, this is superior particle swarm optimization is superior; moreover, the particle swarm optimization is faster compared to the genetic algorithm. Because here we have got no such operators like reproduction crossover and mutation. And moreover, this particle swarm optimization the particles have memory. So, they are little bit intelligent compared to the genetic algorithm. And that is why the particle swarm optimization is expected to perform better compared to the genetic algorithm. But there is a chance of further improvement in each of these particular the algorithms.

Thank you.