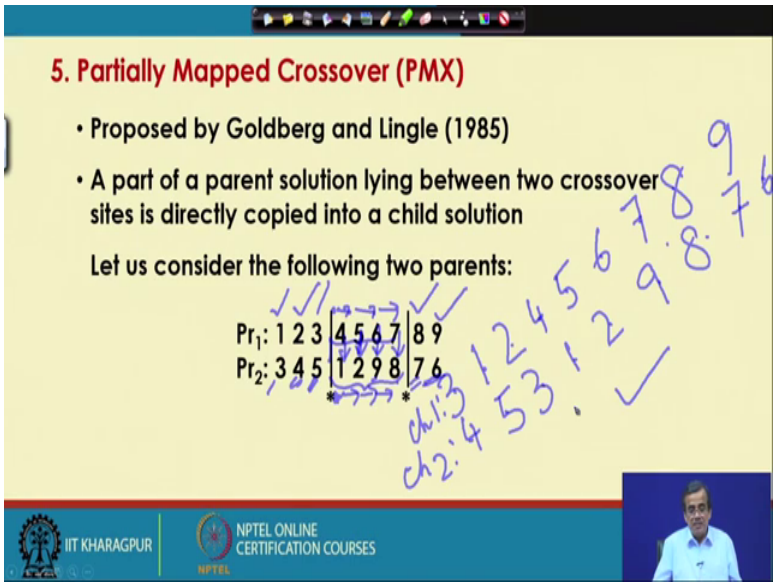
Traditional and Non-Traditional Optimization Tools
Prof. D. K. Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 19
Scheduling GA (Contd.)

Now, we are going to start with the principle of another cross over operator which is known as Partially Mapped cross over PMX.

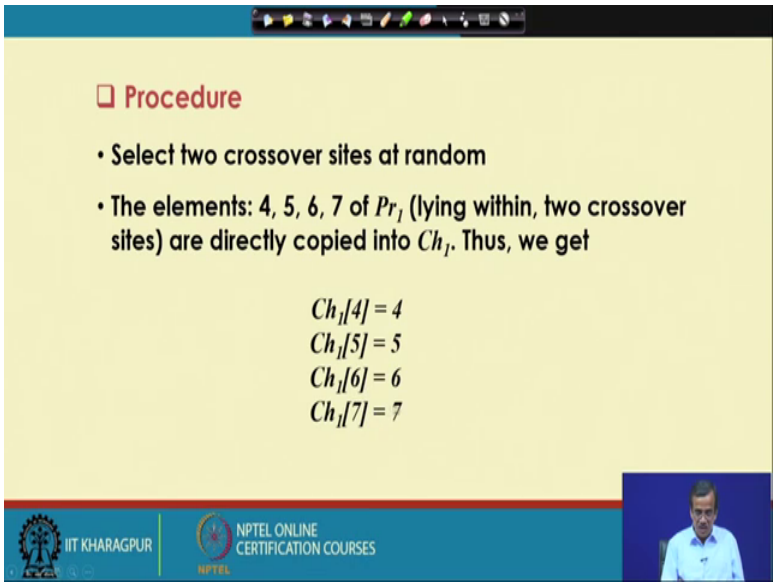
(Refer Slide Time: 00:32)



Now, the concept was proposed by professor Goldberg and Lingle in the year 1985. Now once again we have got the 2 parents, and let us see starting from the 2 parents how to get the valid children's solution. Supposing that once again I am considering the TSP involving 9 cities, and so this is the parent 1, and this is parent 2, and the 2 cross over sides we will have to select at random. So, this is the first cross over site, and this is the second cross over site. Now this principle is slightly with difficult to understand, now what I do is actually I have written all such things in steps.

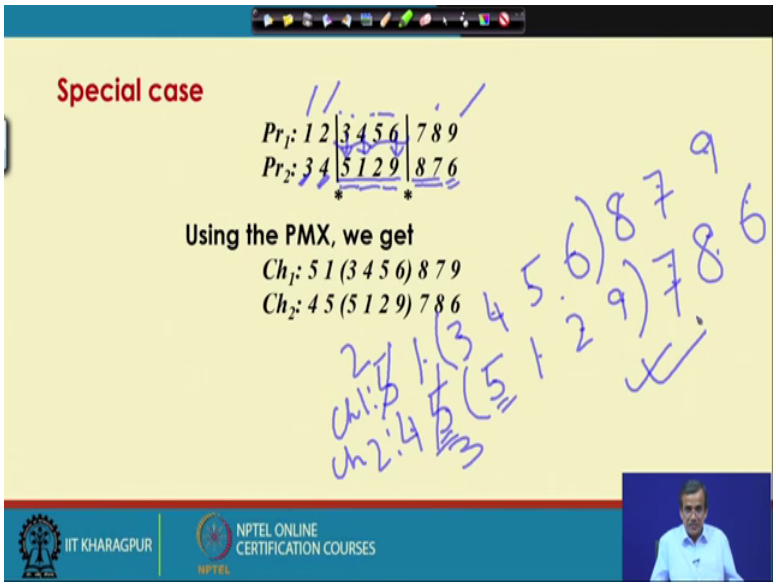
So, I can show you that steps first, then I will be discussing this particular numerical example how to find out the children solution. So, let me first concentrate on like the principle first, let me state that principle first, and then I will be concentrating.

(Refer Slide Time: 01:49)



We select 2 cross over sites at random the elements 4 5 6 and 7 of parent 1 lying within the 2 cross over sites are directly copied into child 1, and thus we get child 1 position 4 is 4, child 1 position 5 is 5 child, 1 position 6 is 6, child 1 position 7 is 7.

(Refer Slide Time: 02:17)



To determine the remaining elements of child 1 the sort starts with the element of parent 1, residing in between the 2 cross over sites. Now let us see what does it mean. So, I am just going to solve that numerical example based on this particular the principle, and I

will show you like how to find out the 2 valid children starting from the 2 parents. Now let me let me try to concentrate on these like how to find out the children solution.

Now, once again my aim is to find out child 1 and child 2. So, let me put child 1, and let me first mark 9 locations 1, 2, 3, 4, 5, 6, 7, 8, 9 and for this particular the child 2 child 2, I will write later on now what I do is first we try to copy, this particular cities lying on parent 1 which are lying in between the 2 cross over sites, like this 4 5 6 and 7 those things are copied fast.

So, so let me let me put 4 here, then comes 5 here, 6 here, and 7 here. Now once I copied the 4 5 6 and 7 from parent 1, now I will concentrate on this particular the city ok, that is your city 4. Now this location I should concentrate first I will try to find out the location of city 4 on parent 2. So, this is the location, and at this particular location I will put a city which corresponds to city 4, but lying on parent 2; that means, this is actually.

So, 1 corresponds to 4 1 is lying on parent 2 4 I have copied from parent 1, and now I am just going to copy something from parent 2, and this particular location of city 4 that is what that is the second location. So, what I do is at the second location I put city 1, next I concentrate on 5 city 5, and I will try to find out the location of city 5 on parent 2.

So, this is the location that is the 3rd location, and at the 3rd location I will have to put city 2, because 2 corresponds to 5 and 2 is lying on parent 2. So, what I do is. So, at location 3, so I will have to put city 2 next I concentrate on 6. So, I will try to find out the location of 6 on parent 2. So, this is the location and 6 corresponds to 9. So, what I do is? .

So, we try to at this particular location I will have to put 9 here, the next is 7 and the location of 7 on parent 2 is this that with this particular location, and 7 corresponds to 8 so, I will have to put 8 here. Now you see there is only a 1 vacant position that is the first position ok, now at the first position I will have to copy directly from you are the parent 2. So, at location 1 on parent 2 so, I have got city 3, and now you can check 1 2 3 4 5 6 7 8 9 so, this is actually a valid child. .

Now what you do is? We try to find out child 2 following the same principle. So, let me first locate 9 locations 1 2 3 4 5 6 7 8 9, now what you do is, so this 1 2 9 8 from parent 2. So, directly you copy keeping their position information intact so, here we have got 1

then comes your 2 then comes 9 8. Now you concentrate here now here we have got city 1, and we will try to find out the location of city 1 on parent 1 so, this is the location, and 1 corresponds to 4.

So, what I do is at location 1 we put city 4. Now we concentrate on 2. So, we try to find out the location of 2 on parent 1. So, this is the location there is a second position and 5 corresponds 2 corresponds to 5. So, at the second position so, I will have to put 5, now I concentrate on 9, so, I will try to find out the location of 9 on parent 1. So, this is the location and 9 correspond to 6. So, at the last position I will have to put 6 next is 8.

Now this is the location of 8 on parent 1. So, 8 corresponds to 7. So, I will have to put 7 here, now you see now the only missing position is actually your the 3rd position, at this 3rd position, I will have to directly take from parent 1 so, on the 3rd position on parent 1 I have got 3. So, let me put city 3 here, and now let me check city 1 2 3 4 5 6 7 8 9. So, all the cities have been considered once there is no missing city missing position and no city has been repeated. So, this is nothing, but a valid child solution. .

So, using the principle on this partially mapped cross over. So, we can find out the 2 valid children from the 2 parents, but remember 1 thing so, using this partial partially mapped cross over sometimes we make it some invalid solution, and if we get some invalid solution, then how to modify it to make it a valid solution that I am going to discuss. Now what I do? I will be taking another example just to show you that we may face some problem using this partially mapped cross over so, let us see that particular example that typical example.

Now, supposing that using the same principle. So, I am solving this particular problem. So, once again I have got 2 parents, parent 1 and parent 2, and these 2 are the cross over site selected at random. Now let us see using the same principle like how to find out like child 1 and child 2. Now let me try to find out the child 1 and child 2.

So, let me write here child 1 child 1. So, 9 location let me first identify 1 2 3 4 5 6 7 8 9, now to get child 1 the first thing I will have to do is so, I am repeating the method once again, so I will concentrate on 3 4 5 6. So, let me directly copy 3 4 5 6. So, 3 I am putting here, 4 5 6. And now I will concentrate here, so I will try to find out the location of city 3 on parent 2 so, this is the location and 3 corresponds to 5 ok. So, for the time being let me put 5 here, although 5 is already there ok, next I concentrate on 4 4 corresponds to 1

and this is the possible location of a 4 on parent 2. So, at this particular position so, I will have to put 1 let me put 1 here. .

Next is 5 so 5 correspond to 2 and this is the possible location of 5 so, at this location I have already put 3. So, I cannot write 2 there ok. So, I am just going to ignore this particular 2. Now I am just going to concentrate on 6, now the possible location of 6 on parent 2 is this and 6 corresponds to 9. So, at the last position I am just put 9, now actually what will have to do is. So, this 3 4 5 6 are taken from parent 1, and now what I am going to do the rest of the thing I am just going to take from the parent 2, now the vacant positions are 8 and 7. So, this actually your the vacant position.

So, let me put 8 here and 7 here so, this 8 and 7 I am copying it here ok. So, according to the principle of this PMX this should be my child 1, now let me check you see city 1 has come once, but 2 is missing 3 has come 4 has come 5 6 7 8 9, but 5 has come twice ok. Now if you remember the 4 cities which all lying between the 2 cross over site, which we copied first are nothing, but this.

So, this 4 we copied first and whatever we are putting inside the first bracket we do not want to change, now what we do is as I told the 5 has come twice, but this 5 we do not want to change, but here I am got another 5, now this 5 I can replace by the missing city that is city 2. So, what I do is I replace this 5 by 2 because. otherwise this child 1 will become an invalid child to make it valid what you do is. So, 5 will be replaced by the missing city that is city 2.

And now you see like 1, 2, 3, 4, 5, 6, 7, 8, 9 so all the cities have come once there is no missing city no missing place no city has been repeated. So, this is a invalid child. So, this is that way actually will have to make the correction, to make to convert the invalid child to a valid 1. Now let us see how to find out child 2 using the same principle. So, let me locate 9 positions 1, 2, 3, 4, 5, 6, 7, 8, 9 and let me copy this 5, 1, 2, 9 first so, this 5 1 so this is 5 1 2 9 and let me put under the first bracket.

Now you concentrate on the city 5 lying on parent 2, now the possible location of city 5 on parent 1 is here ; that means, at this particular position I have already copied something that I do not want to disturb. So, let me ignore this next is 1, this is the possible location of 1 and parent 1, and 1 corresponds to 4. So, at first position you put 4

here the next is 2, 2 corresponds to 5, and the possible location of 2 on parent 1 is this the second position. So, second position for the time being you put 5 ok.

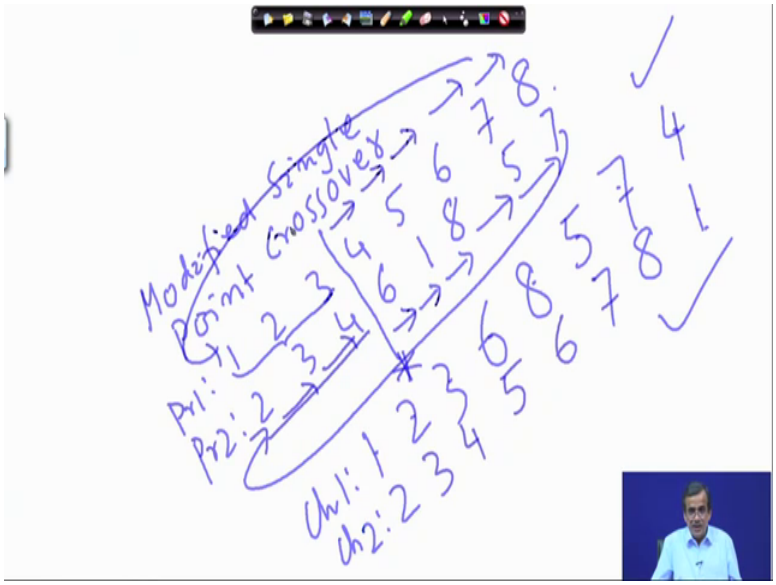
The next is 9. So, this is the location of 9 on parent 1 and 9 corresponds to 6. So, at this particular position that is the last position. So, I will have to put 6 here, and the vacant position. So, this and this are to be copied from this particular the parent 1 that is I will have to write 7 here, and 8 here are you getting my point. So, using this I can find out the children solution that is child 2.

But let me check the validity, now city 1 is here city two, but there is missing 4 is here, but 5 have come twice, 6 is here, 7 8 9 all are there ok, the only thing is your the 5 has come twice. Now as I told the cities which are lying within the first bracket we are not going to change, and the city which is lying outside the bracket, and if it is repeated then that can be replaced by the missing city. Now here 3 is the missing city. So, in place of 5 let me put 3 here. So, the valid child that is child 2 is your 1, 2, 3, 4, 5, 6, 7, 8, 9 so, all the cities have come and this will be your valid child 2. So, using this particular correction so, I can find out the valid children starting from the 2 parents.

But this particular problem we are not going to phase always and sometimes we may face problem and if you face this type of a invalid children. So, will have to make it valid by making this type of correction, then only this will become valid children, this is actually a I should say a debater little bit of this particular PMX that is the partially mapped cross over. .

Now actually I am just going to tell that we have already discussed while discussing a the principle of binary coded chain that single point cross over, now the concept of single point cross over can also be used to design 1, special cross over operator for this type of scheduling problem. Now let me let me try to try to concentrate on that. So, what I am going to do is actually I am just going to discuss a modified single point cross over.

(Refer Slide Time: 18:17)



Now the modified so, modified single point cross over let me try to design, modified single point cross over.

Now, let me take the 2 parents at random say parent 1, let me take 1, 2, 3, 4, 5, 6, 7, 8 say there are only 8 cities, now parent 2 let me take at random like 1, 2, 3, 4, 5, 6, 7, 8. So, parent 1 is 1, 2, 3, 4, 5, 6, 7, 8 and parent 2 is 2, 3, 4, 6, 1, 8, 5, 7 so, these are taken at random. Now let us see how to implement a slightly modified version of single point cross over, which we discussed long back while discussing the binary coded g a to solve this type of problem.

Now what we do is just like the single point cross over like, we select the cross over site only 1 cross over site at random, supposing that my cross over site is here, only 1 cross over site. Now according to the principle of this single point cross over the elements which are lying on the left hand side of the cross site will remain same, and the elements which are lying on the right hand side there will be stopping.

Now, let us let us try to use the same principle here. So, there are 8 locations 1, 2, 3, 4, 5, 6, 7, 8 and as I told that thus so, to get the child 1, the cities which are lying on the left hand side of the cross over site let me copy first 1, 2, 3 so 1, 2, 3, then starting from here I am just going to take the information from parent 2. And let me start from here, now if I start from here so, the first 1 is city 6 6 has not come so, let me put 6 here. Next is 1, 1 is already there so I cannot retain next is 8, 8 is not there so I can take it, next is 5; 5 is also

not there so, I can consider 5. Next is 7, 7 is also not here so, I can take it now there is 1 vacant position. Now to fill up this particular vacancy actually what you do is we come back here 2, 2 is already there, 3 is already there, 4 is not here. So, let me put 4 here now you check whether all the cities have come once or not city 1, 2, 3, 4, 5, 6, 7, 8. So, this is actually a valid child, now this is slightly modified version of the single point cross over, but will be getting the valid child here also. .

Then child 2 to get the child 2 so, directly so, 2 3 4 you copy so let me copy here like 2 3 4, and then let me start my search from here the city 4 is already considered, then comes city 5 5 is not here so, let me put 5 here, next is city 6. So, let me put 6 here, next is city 7 7 is not there so, let me put 7 here, next is city 8. So, let me put 8 here, then there is 1 vacant position so let me come back.

So, city 1 one is not there so let me put city 1 now you check city 1, 2, 3, 4, 5, 6, 7, 8. So, all the cities have come only once. So, this is actually a valid child. So, using this slightly modified single point cross over, we can also find out the valid children from the 2 parents.

Now, till now we have discussed a few cross over operators these are all special type of cross over operators to be used for scheduling g a. Now my question is out of all the cross over operator which 1 is the best, in fact what I did we carried out some experiment actually all the cross over operators, in fact we tried with a number of test function, and we will see that that the performance of these particular cross over operators is function dependent.

Thank you.