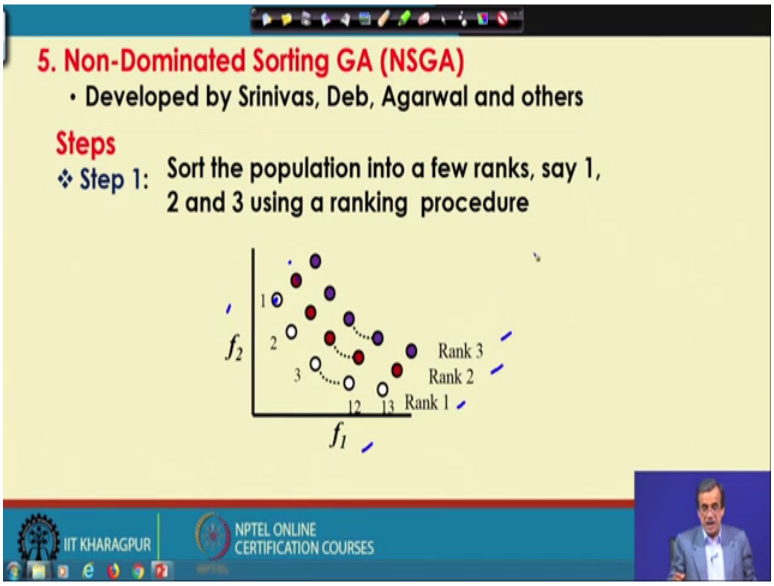
Traditional and Non-Traditional Optimization Tools
Prof. D. K. Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 25
Multi-Objective Optimization (Contd.)

We are going to start with the working principle of another very popular tool for multi-objective optimization and this is known as non dominated sorting genetic algorithm; in short NSGA.

(Refer Slide Time: 00:32)



Now the initial idea of this NSGA came from pressure David Goldberg and the first version was developed by Srinivas Deb and then and later on, it is further modified by Agarwal Dave and others. Now let us try to see the working principle of this particular the non dominated sorting genetic algorithm.

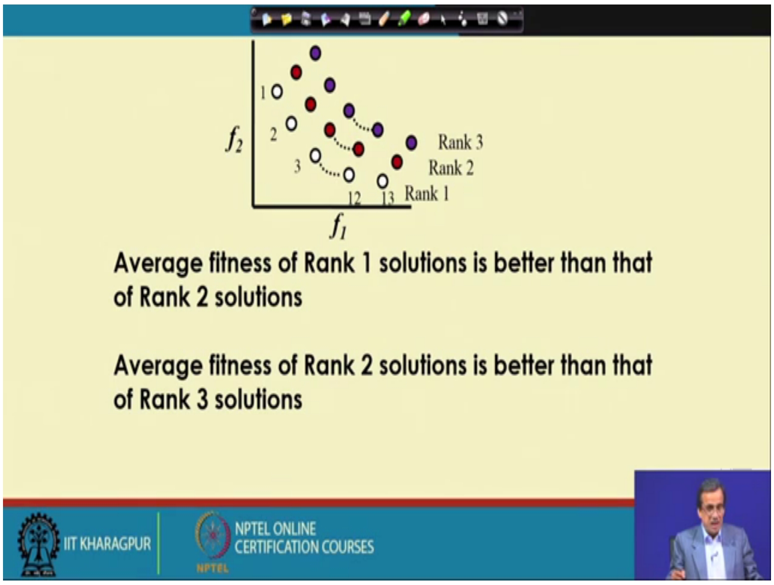
Now, let us start with step one we sort the population into a few ranks say one 2 3 and so on. Now as we discussed that NSGA starts with a population of solution selected at random and what we do is the first we try to sort them into a few ranks based on their fitness values. So, we compare the fitness values of 2 solution and try to find out which one is a non dominating one now the rank 1 fraud consists of all non dominated solution. So, the whole population can be divided into a few ranks like rank 1, rank 2, rank 3 and so on. Now I have already discussed, how to do this particular the ranking. Now, once you have got this particular rank, now we can proceed with like the first operator that is reproduction and the purpose of reproduction is to select the best solution probabilistically just to form the mating pool. Now, supposing let me concentrate on this particular problem. Now here, we are going to minimize 2 objectives say f 1 and f 2 and supposing that from the initial population selected at random. So, we are able to sort into a few ranks like your rank 1 rank to rank 3 and so on.

Now, if I compare a particular solution from rank 1 and another solution from rank 2 definitely. So, this particular rank 1 solution is non dominated with comparison with your the rank 2 solution; that means, this particular solution the rank 1 solution is slightly better compared to this particular solution in terms of the objective function. Now, if we consider the average fitness of rank 1 solution, average fitness of rank 2 solution and average fitness of rank 3 solution, now it indicates that the average fitness of rank 1 solution will be better compared to that of rank 2 and the average fitness of rank 2 solution will be better compared to that of rank 3 solution and so on.

Now, if I just go for the reproduction scheme just to get the mating pool there is a possibility that in the mating pool there will be more number of solutions belonging to rank 1 and so, this particular number of solutions belonging to rank 1 is going to increase in the population; that means, there will be some selection pressure and gradually what will happen is.

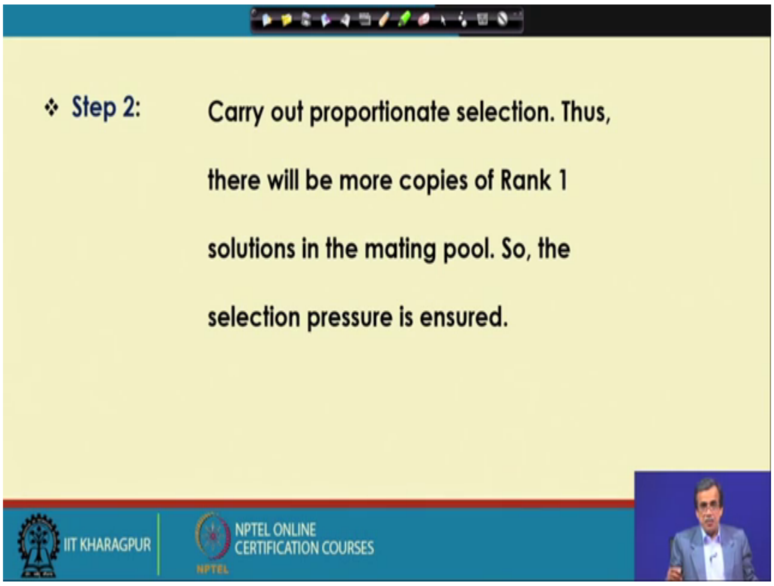
So, the mating pool will consist of the more number of better solution and consequently j will try towards determining the optimal or the globally optimal solution. So, in this way actually we can ensure the selection pressure, but as we know that if I want to ensure a very efficient search for the j. So, we will have to maintain a proper balance between the selection pressure and population diversity. Now I have already discussed little bit how to ensure this particular selection pressure by selecting more number of rank 1 solution in the mating pool..

(Refer Slide Time: 05:07)



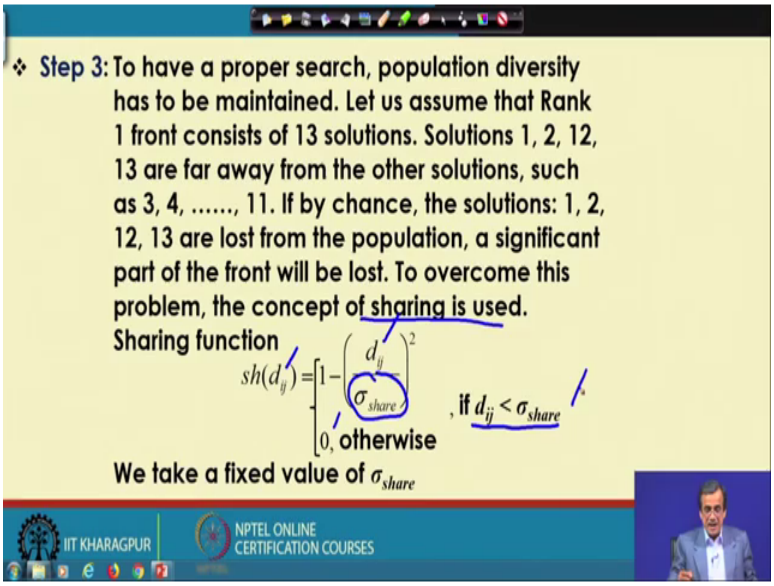
Now let us see whatever I told here. So, those things I have written it that the average fitness of rank 1 solution is better than that of rank 2 solution. Similarly the average fitness of rank 2 solution is better than top rank 3 solution and we carry out proportionate selection.

(Refer Slide Time: 05:23)



Now here in proportionate selection which we have already discussed the probability of selection is proportional to the fitness. So, there is a possibility in the mating pool there will be more number of rank 1 solution and which is going to ensure the selection pressure in this particular the GA search.

(Refer Slide Time: 05:47)



Now, let us see how to maintain the diversity now as we have already discussed several time that to ensure a very efficient search for the GA, a proper balance between selection pressure and population diversity has to be maintained selection pressure we have already discussed. Now, I am just going to concentrate how to get or how to ensure the population diversity. Now, let us see how to ensure the population diversity. Now let me for the time being concentrate on the rank 1 solution. Now if we see the earlier figure. Now, we can see that in the rank 1. So, there are 13 solutions. Now, out of these 13 solutions, we can see that solutions 1, 2 and say 12 and 13, they are far from the other solution.

Now, here there could be a very close solution like 3 4 5 6 and all such things they are very close, but these 2 solutions one and 2 and these 2 solutions 12 and 13 they are too far from the other solution. So, I am just going to consider there are 13 solutions on the rank 1 and out of 13, 1 in 2 and then 12 and 13, they are too far from the remaining solution; now this is the problem.

Now if this is the problem, then there is a possibility that in the next iteration or to the next to next iteration, the end solution like 1 to 12 and 13 will be deleted from the population and there is a possibility we may not get the whole of the stretch of this particular the Pareto optimal front of solution; that means, a part of the Pareto optimal front could be lost and if it is lost then the diversity from the GA search. So, that will be lost and that is not a very efficient search. Now, how to ensure how to how to ensure that all the solutions would be there in this particular Pareto optimal front and the stage of the Pareto optimal front will be maintained.

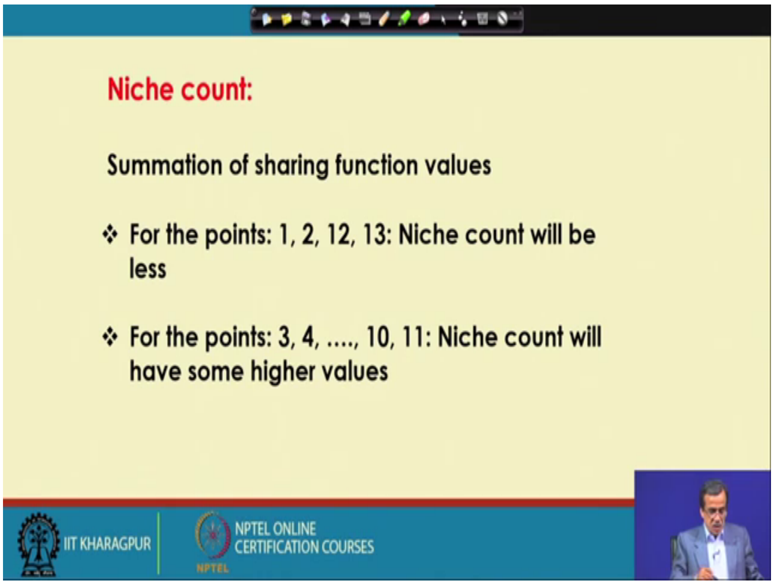
Now, to see this particular thing how to maintain this particular diversity; so, we take the help of this step 3. Now here actually what we do. So, we have considered 13 solution in the rank 1 and as I told solutions 1 2 12 and 13 are too far from the other solution and there is a possibility those solutions will be lost; that means, a significant past will the Pareto optimal front we will lost from the Gf solution.

Now to overcome this particular problem we use the concept of sharing. Now let us see how to use this particular the sharing now to implement the concept of this particular sharing function we take the help of the equilibrium distance. So, we try to calculate the equilibrium distance between the 2 points; i and j that is denoted by d ij and once we know this particular equilibrium distance using this particular expression.

So, we can find out what should be the sharing between the 2 points i and j that is denoted by s h di j. So, the sharing between 2 points d ij is nothing, but Sh di j and that is nothing, but one minus di j divided by sigma share square if di j is less than sigma share otherwise. So, this particular sharing between the 2 points i and j will be assumed to be equal to 0. Now here this particular sigma share is a fixed numerical value and that is actually assigned by the user how to assign this particular sigma share value it depends on the experience of the user and of course, it depends on the problem information.

Now, this I am going to discuss up to some time in more detail, but for the time being let me see if I know the distance between the 2 points i and j. So, how to determine; so, this particular the sharing function value now here I just want to mention if the distance is less than sigma share the predefined fixed value then only. So, I will be getting some non0 value for this particular the sharing, but if the distance is greater than equals to sigma share; that means, the 2 points are too far, then the sharing between them will be equal to 0. Now using this particular principle and if I take the same example now.

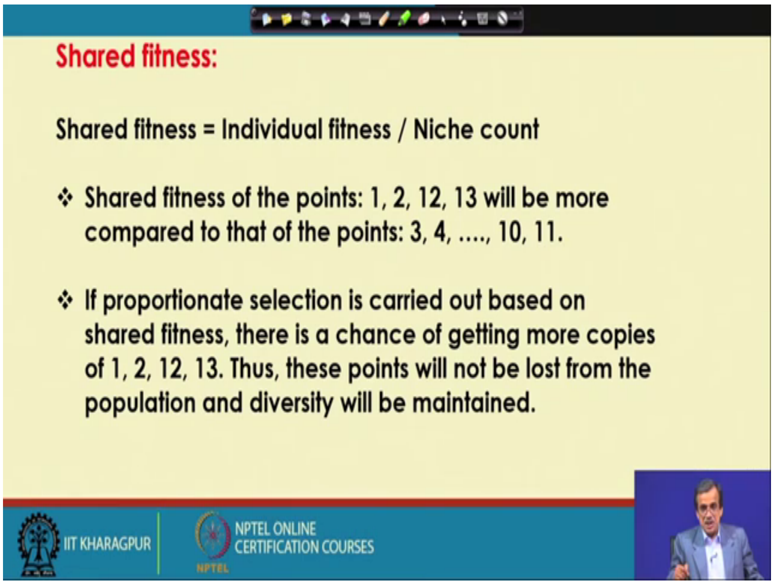
(Refer Slide Time: 11:00)



I can come to the conclusion that the points 1, 2, 12 and 13 the sharing value the sharing function value will be very small on the other hand for the remaining points like your 3, 4, 5, 6 and all such things all such points up to 11 the sharing function value will be somewhat on higher side. Now what I do is we try to calculate the niche count. Now the niche count is nothing, but the summation of the sharing function values.

Now, if I assume that there are 13 points on the rank 1. So, what I do is for each of the 13 points, we try to find out what is the niche count value and if I just find out the niche count value will be getting the information that the niche count for the points 1, 2, 12 and 13 which are too far from the other points the niche count will be less and the points which are lying very close to each other that is 3, 4, 5, 6 up to say 11 the niche count value will be somewhat higher. So, the same thing I have written it here for the points 3, 4, 10 up to 11 the niche count will have some higher values..

(Refer Slide Time: 12:29)



Now what we do is we try to calculate the shared fitness. Now the sad fitness is nothing, but the individual fitness divided by the niche count and we have already discussed the niche count value for the points 1, 2, 12 and 13 will be will be less; that means, the shared fitness for the points 1, 2, 12 and 13 will be more compared to that of the points 3, 4, 10 and 11.

Now, let me summarize whatever I discuss. So, what I do is supposing that there are 13 points on the rank 1 front. So, for each point I try to find out. So, what will be that niche count. Now on niche count is nothing, but the summation of the sharing function values and to determine the sharing function values we take the help of the Euclidean distance.

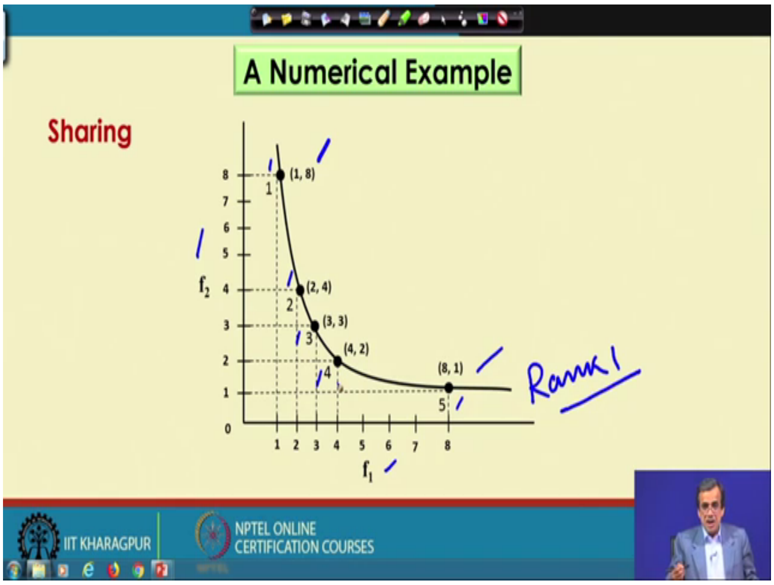
Now if the points are too far, then the Euclidean distance will be more; that means, you are the sharing function will be less and if it exceeds, if the distance between the 2 points exceed a certain predefined value the sharing between these 2 points will be 0; that means, the points which are lying on the end the 2 ends which are too much distant from the other points the sharing function value will be less and consequently the niche count will be less and that is why for the point. So, 1, 2, 12, 13 the niche count value will be less and the shared function shared fitness is defined as individual fitness divided by niche count.

Now, what it is the individual fitness initially we assume that they are equal and the niche count value for the point 1, 2, 12, and 13 will be more will be less the niche count value will be less compared to the other points like 3, 4, 10 and 11 consequently the shared fitness of the points 1, 2, 12 and 13 will be more compared to that of the points 3, 4, 10 up to 11. Now if I do the reproduction or the proportionate selection based on the shared fitness in place of the original fitness; so, there is a possibility that in the mating pool there will be more copies of these particular points like 1, 2, 12 and 13.

Now, if I get more copies of the points 1, 2, 12 and 13 in the mating pool. So, this mating pool is going to participate in crossover and after that there will be mutation and there is a possibility that in the mating pool as we have considered good solution. So, in the will be getting slightly better children solution and by doing that the population diversity will be maintained for this particular the GA population.

Now this is the way actually we can incorporate or we can maintain the population diversity in the GA. So, this is how to maintain that particular the population diversity in the GA now whatever I discuss the same thing I have written it here if proportionate selection is carried out based on the shared fitness there is a chance of getting more copies of 1, 2, 12 and 13 the these points will not be lost from the Pareto upfront and diversity will be maintained. So, this is the way actually we ensure the population diversity.

(Refer Slide Time: 16:21)

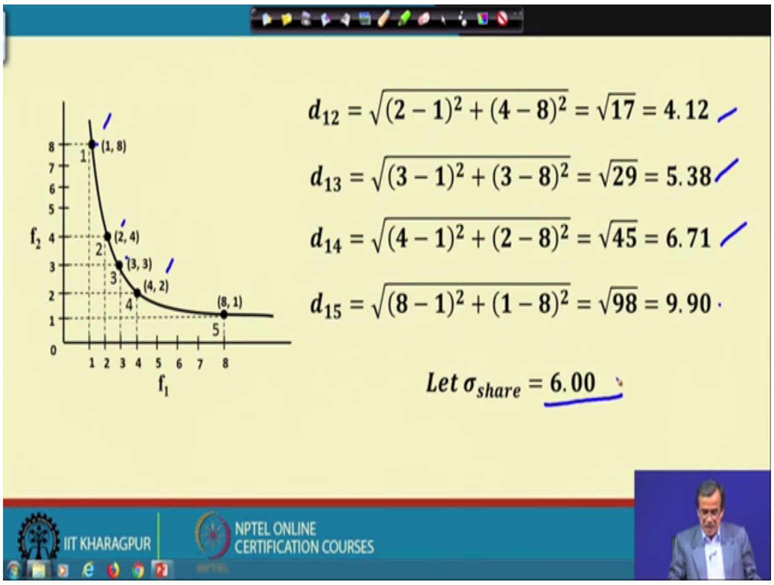


Now, we have seen that we are able to maintain the selection pressure as well as the population diversity the method I have already discussed now once we are able to maintain a balance between selection pressure and population diversity now we are in a position we are in a position to improve the solution further with the help of the operators like crossover and mutation now after this reproduction scheme is implemented now we go for the crossover and the mutation and that completes actually one iteration or one generation of the GA.

Now, this process will go on and go on through a large number of iteration and ultimately we will be getting Pareto optimal front of solutions now before I end here. So, let me solve a numerical example just to just to show you the working principle of this particular the sharing function how does it work how to calculate this particular the niche count value and how to find out the shared fitness values. Now let me let me just try to concentrate on this small numerical example. Now the same problem, I am just going to discuss I will have to minimize the 2 objective functions f 1 and f 2 and supposing that this is actually the rank 1 front.

So, this is the rank 1 front and let me assume that there are lying only 5 solutions on this particular the rank 1 front say solution 1, 2, 3, 4 and 5 and supposing that these are the coordinates of these particular the 5 solution; that means, corresponding to solution 4 now the value of f 1 is for at a value of f 2 is to 2 and this is denoted by 4 comma 2. Similarly, I can find out the coordinate of all such points and a particular the coordinate of a particular point indicates how much is f 1 and how much is f 2 and supposing that we have got only the 5 points here now out of these 5 points 0.1 and point 5 are too far from the other points like say 2, 3 and 4 and let us see how to implement the concept of this particular the shared fitness how to calculate that.

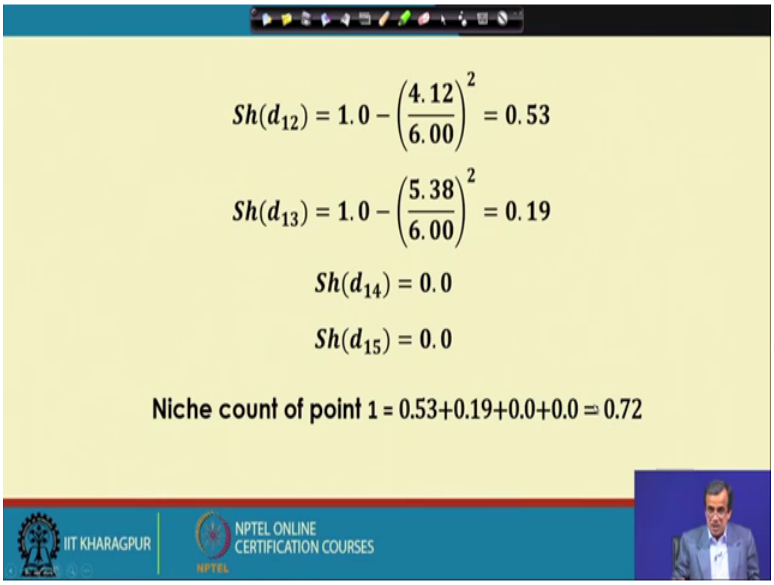
(Refer Slide Time: 19:06)



Now, to calculate the niche count for this particular 0.1. So, what we do is we try to find out the equilibrium distance between 1 and 2 and that is nothing, but square root 2 minus 1 square plus 4 minus 8 square that is nothing, but square root 17 that is 4.12. Similarly we can find out the distance between 0.1 and 3 that is d 1 3 and exactly in the same way, I can find out the Euclidean distant 5.38 next is we try to find out the Euclidean distance between 1 and 4 and following the same principle, I will be getting 6.71. Next is the Euclidean distance within 1 and 5 and that is nothing, but this and I will be getting 9.90. So, these are all Euclidean distance values calculated with respect to the 0.1..

Now actually what we do is we will have to set some fixed value for this particular sigma share and let me assume that sigma share is equal to 6.00. Now if this is the situation, then now let us see how to find out.

(Refer Slide Time: 20:27)



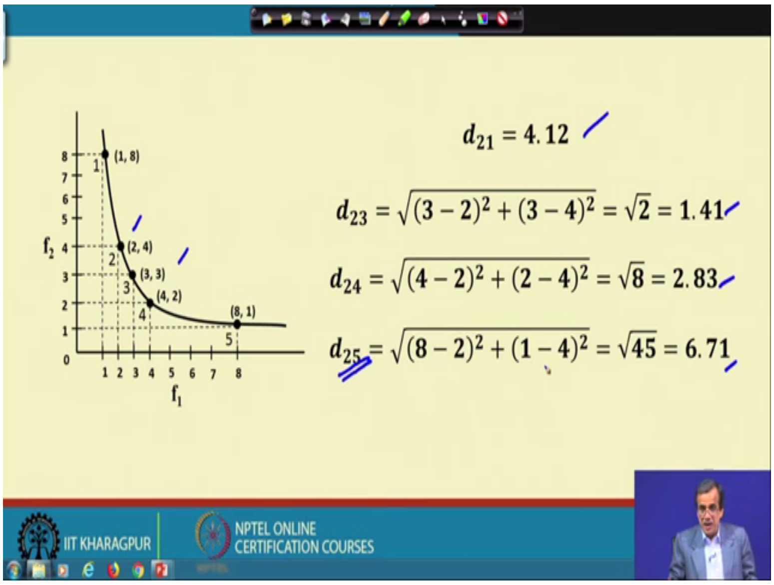
The sharing function the sharing function d 1 2 that is the sharing function value between 1 and 2 is nothing, but 1 minus 0 minus 4.12 divided by sigma share that is 6 4 0 0 square because here the d ij is 4.12 and that is less than the sigma share that is 6.10 and that is why. So, we will have to use this particular expression to calculate the sharing between one and 2 and this will become equal to 0.53.

Similarly, the sharing between 1 and 3 that is Sh d 1 3 is nothing, but 1.0 minus 5.38 divided by 6.00 square and here, once again the Euclidean distance between 1 and 3 is less than 6.00. So, we will have to use this particular expression and this will become equal to 0.19, next is the sharing between the 1 and 4 the distance between 1 and 4 that is d 1 4 is found to be more than 6.00.

So, the sharing between 1 and 4 will become equal to 0.0. Similarly the sharing between 1 and 5 will be 0.0 as the Euclidean distance between 1 and 5 is more than 6.00. Now if this is the situation now very easily I can find out what should be your niche count. So, the niche count for 0.1 is nothing, but 0.53 plus 0.19 plus 0.0 plus 0.0. So, this will become 0.72.

So, for the niche count for this particular 0.1 is nothing, but 0.72..

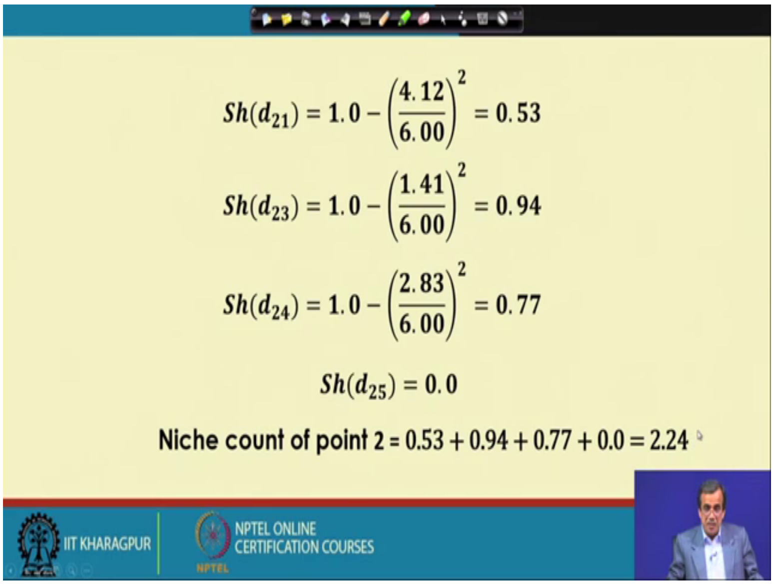
(Refer Slide Time: 22:36)



Now next we try to find out the niche count for the second point; that means, this particular point. So, we try to find out the Euclidean distance between 2 and 1. So, this is the distance between 2 and 1 the Euclidean distance between 2 and 3. So, this is the distance between 2 and 3 next is 2 and 4. So, this is 2 and 4; 2.83, then 2 and 5 and this is becoming equal to 6.71. Now if I compare the Euclidean distance values with the sigma share sigma share is assumed to be equal to 6.00. So, for this particular point the Euclidean distance is more than your sigma share.

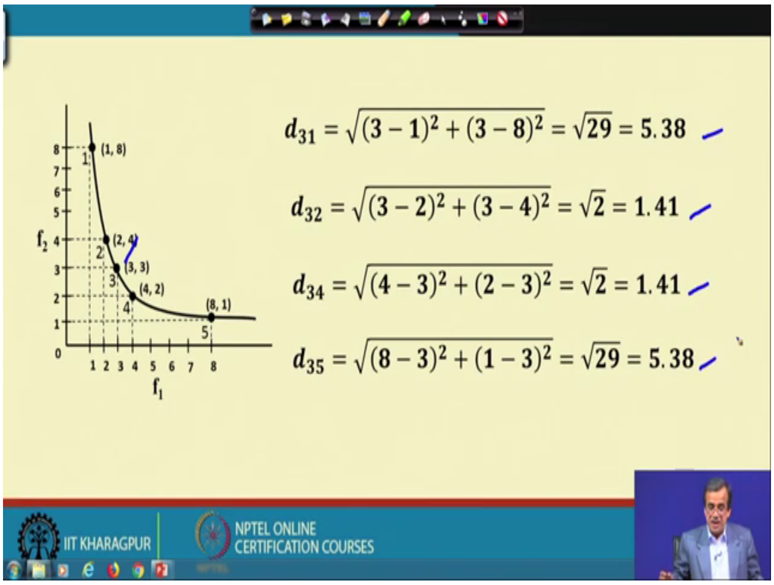
So, its contribution towards the sharing function value will be equal to 0.0. Now if I follow the same principle to calculate.

(Refer Slide Time: 23:34)



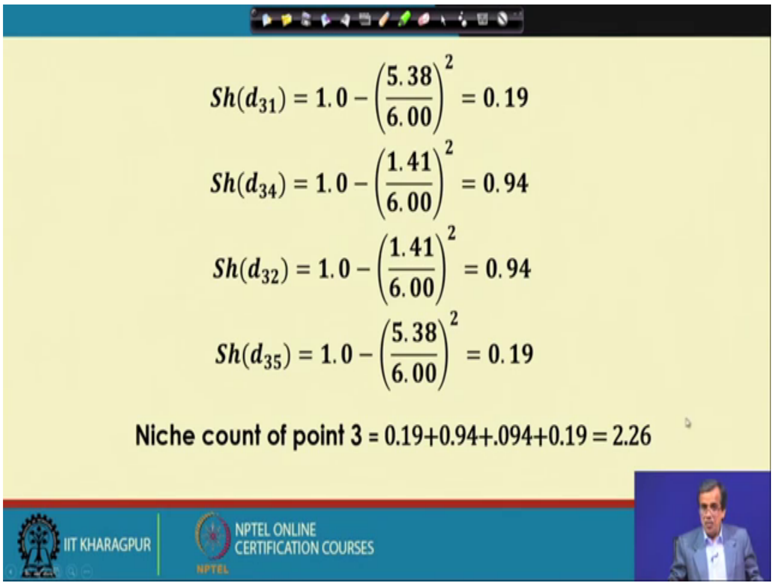
The sharing function value the sharing between d 2 1 is nothing, but 1 point is 0 minus 4.12 divided by 6.00 square that is 0.53. Next is the sharing between d 2 3 is nothing, but 0.94 the sharing between d 2 4 is nothing, but 0.77 and the sharing between d 2 5 is equal to 0.0. Now I can find out the niche count for point 2 that is nothing, but 0.53 plus 0.94 plus 0.77 plus 0.4; 0.0. So, this will become equal to 2.24. So, this is the niche count for the 0.2. Similarly I can find out the niche count for this particular the 0.3.

(Refer Slide Time: 24:28)



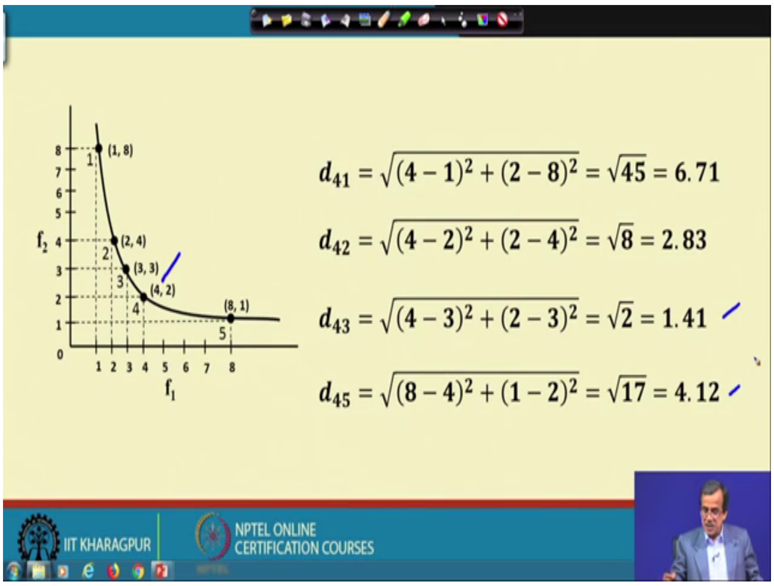
So, I can find out all the distance values like d 3 1 is 5.38, d 3 2 is 1.41, d 3 4 1.41, d 3 5 5.3 and very easily I can find out the niche count value.

(Refer Slide Time: 24:53)



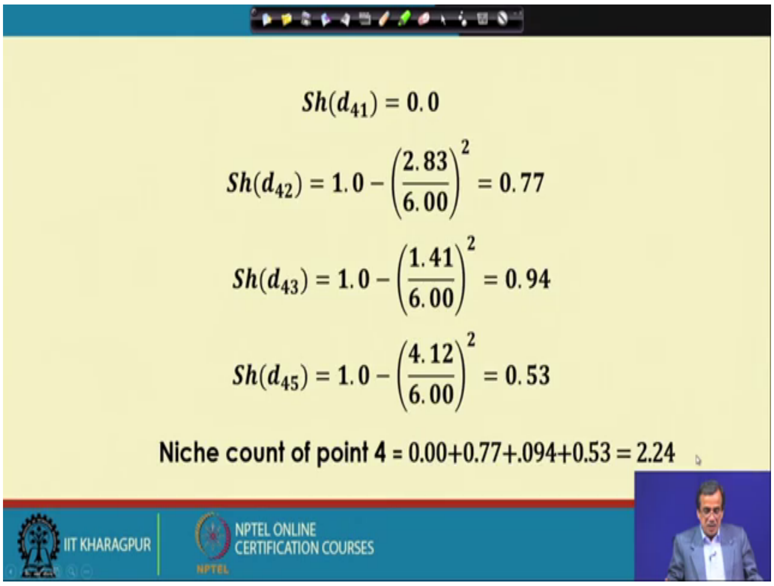
So, the sharing function value; I can calculate between 3 and 1 is 0.91 between 3 and 4, it is 0.94 the sharing between 3 and 2 is 0.94 the sharing between 3 and 5 is 0.19 and now I can calculate the niche count for 0.3 and if we just add them all such values you will be getting 2.26. So, this is the niche count value for 0.3. Now I see how to find out the niche count value for 0.4.

(Refer Slide Time: 25:23)



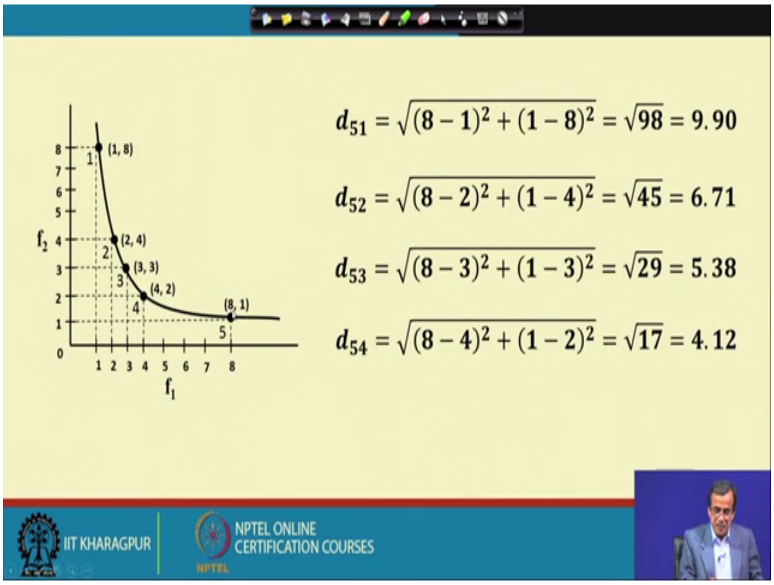
So, the distance between 4 and 1; that means, I am trying to find out the niche count for 0.4. So, we consider d 4 1; 6.71, d 4 2; 2.83, d 4 3 is 1.41, d 4 5 is 4.12 and here you can see that only this d 4 1 is more than 6.00. So, its contribution towards the sharing function value calculation will be 0 and others will have some non 0 value and if I calculate.

(Refer Slide Time: 26:09)



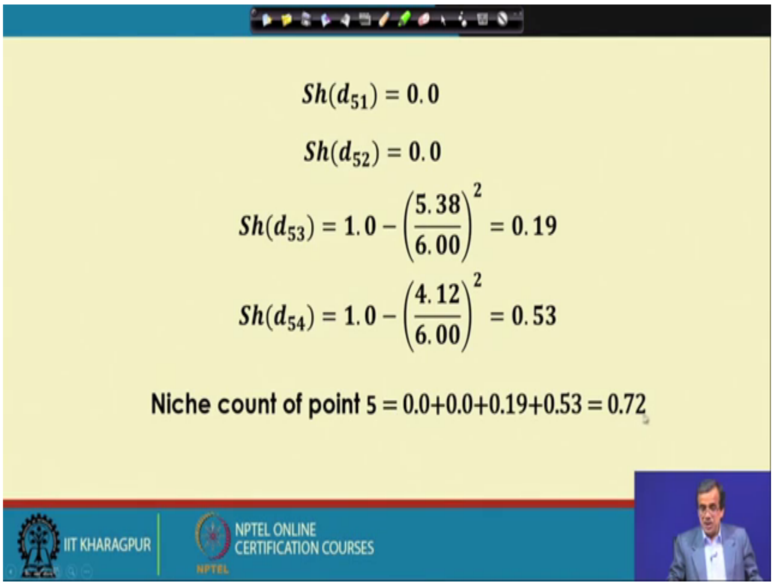
So, I will be getting the niche count the sharing function values are like this and the needs count for 0.4 is 2.24. Now let us see how to find out the niche count for the fifth point. So, this is the fifth point.

(Refer Slide Time: 26:21)



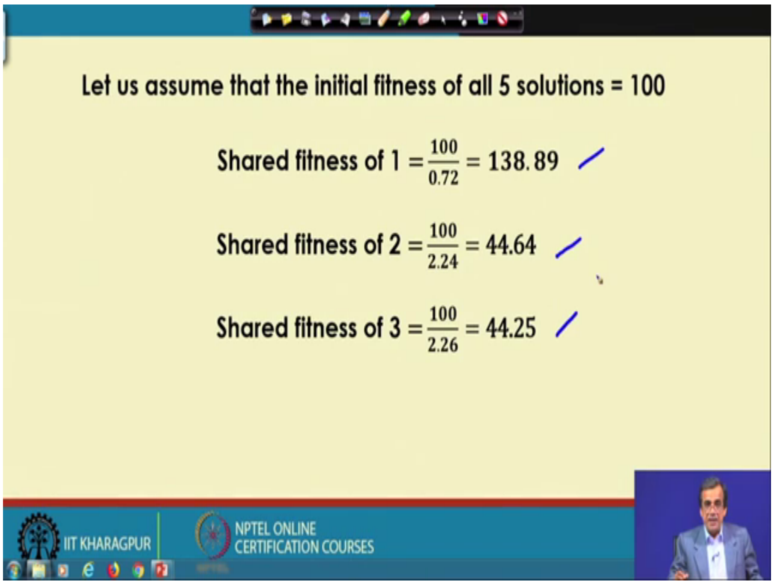
So, once again we determine the Euclidean distance between 5 and 1 that is 9.90 between 5 and 2 6.71 between 5 and 35.38 between 5 and 4; 4.12 and here if I see the equilibrium distance between 5 1 and 5 2 are more than 6.0. So, their contributions towards the sharing function value will be 0 and others will have some non 0 value..

(Refer Slide Time: 27:00)



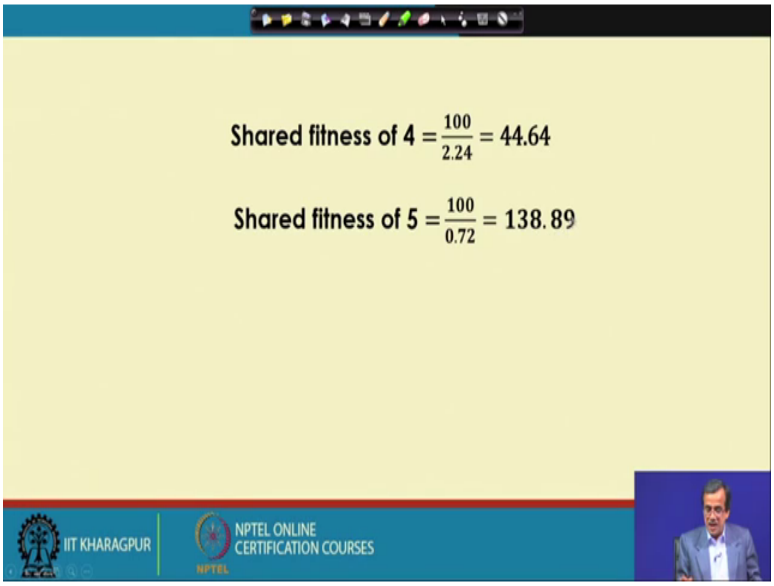
Now if this is the situation. So, very easily I can find out how much is the sharing function value for the different of between 5 1, 5 2, 5 3 and 5 4 and if we just add them up we will be getting the niche count for 0.5 and this is becoming equal to 0.72.

(Refer Slide Time: 27:22)



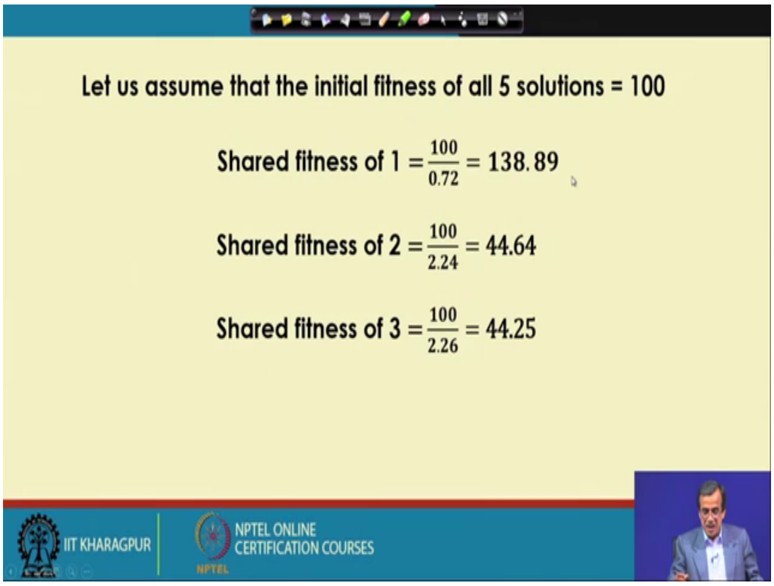
Now, we have calculated the niche count value for all the 5 points which are lying on the rank 1 front. Now let us assume that the initial fitness for all 5 solutions is equal to 100. Now if this is the situation, now I can find out the shared fitness of 0.1 that is nothing, but one hundred divided by 0.72 is the niche count and this will become equal to 138.89. Similarly, the shared fitness value of 0.2 is 100 divided by 2.24 is 44.6 for the shared fitness of the 0.3 is 100 divided by 2.26 is 44.25 and similarly I can find out the shared fitness value for the other points.

(Refer Slide Time: 28:13)



Like that is for the fourth point this will become 44.64 and the shared fitness of the fifth point is hundred divided by 0.72 is 138.89.

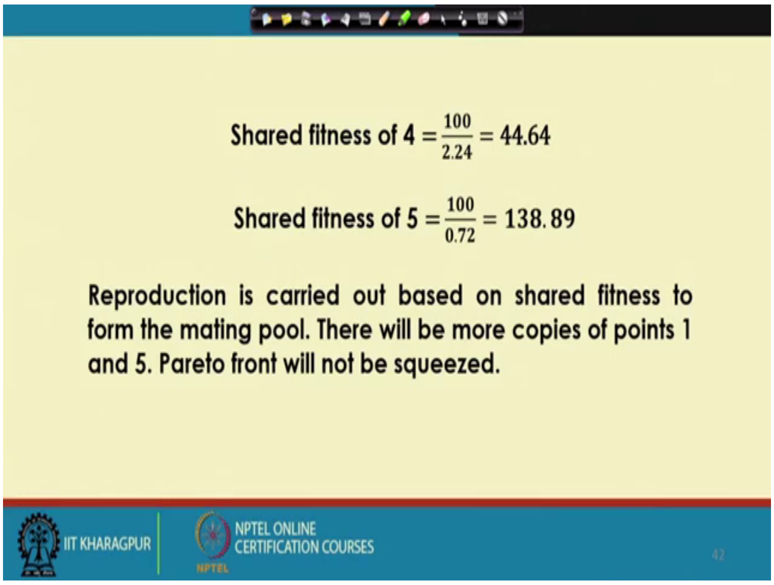
(Refer Slide Time: 28:40)



Now, if I compare the shared fitness values the shared fitness values for the fifth point and for the first point is somewhat higher compared to that of the other points. Now if I do the proportionate selection based on the shared fitness there is a possibility that I will be getting more number of copies of 0.1 and 0.5 in the mating pool and if I get more copies of this 0.1 and 0.5, there is a possibility that this particular end information of this your; the Pareto front that is 1 and 5 will not be lost from the population and this particular the diversity or the spread of this particular front will be maintained and there is a possibility the GA is going to find out a very wide Pareto optimal front of solution.

Now, this is the way actually this concept of sharing works; that means, how to ensure actually this particular the diversity in the population of this particular the GA.

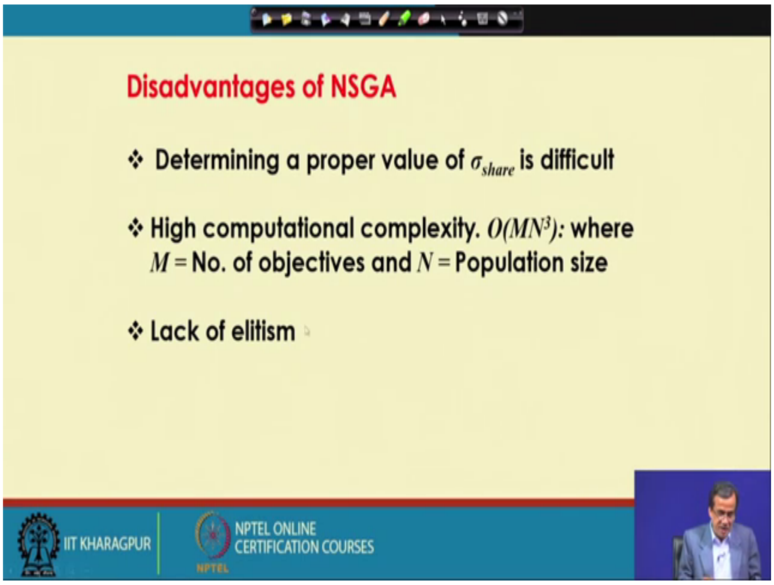
(Refer Slide Time: 29:43)



And once you have done this particular reproduction scheme we got the mating pool and now we will go for the crossover and mutation that completes one cycle of the GA and you will be getting the better and better solution and GA will be able to find out what should be the global Pareto optimal front of solution.

Now, this NSGA could achieve some success and popularity, but it has got a few drawbacks. Now let us try to mention those particular the demerits.

(Refer Slide Time: 30:18)



The demerits are such like here the performance depends on this particular the sigma share value now it is user-defined now how to set this particular the sigma share value whether it is 6.00 or 7.00 that depends on the problem and the user will have to define this particular the numerical value and the quality of the solution will be controlled by this particular the sigma share.

So, determining a proper value for this sigma share is a difficult task the next is your the computational complexity now this particular algorithm that NSGA is computationally complex and the computational complexity is in the order of M nq her M is nothing, but the number of objectives say it is either 2 or 3 and capital N is nothing, but the population size.

So, this particular algorithm is actually the computationally the slower and its computational complexity is more and more over in this NSGA the concept of elitism was not used now I have already discussed the concept of elitism like if I get a good solution I will have a direct copy of this particular good solution in the next in the next population and; that means, the already found good solution we do not want to lose and as I told that this is almost similar to the fixed deposit we do not want to touch and we want to keep a copy of this particular good solution in the population that is the concept of elitism and this concept of elitism is actually not used in this particular the NSGA these are all drawbacks of this particular NSGA.

Now how to overcome how to overcome this particular the demerits of NSGA; that I am going to discuss.

Thank you.