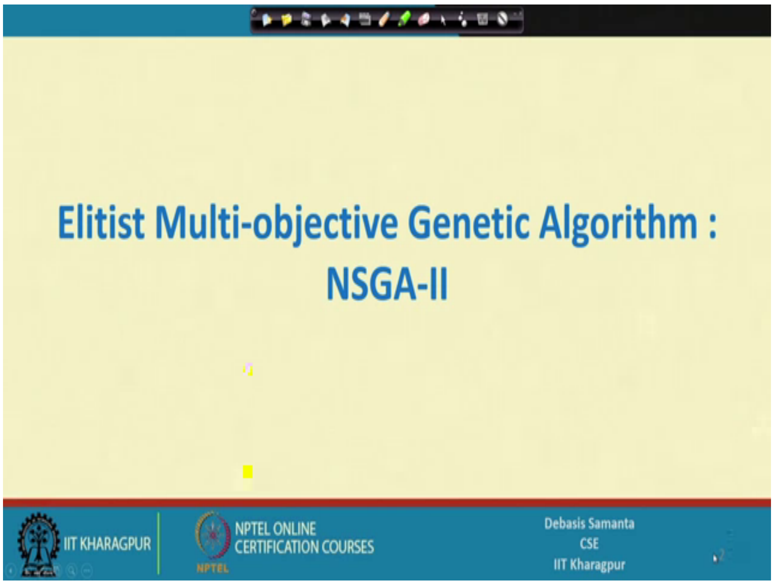
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 33
Pareto-based approach to solve MOOPs (Contd.)

We are discussing NSGA 2 approach, and the NSGA 2 approach followed as some method similar to the NSGA, and it is basically the first method. First step that is they are in both common in a NSGA 2 are non-dominated sorting, front calculation based on the non-dominate sorting or a procedure. Now today and then the next procedure that is the here is different is basically the selection for the mating pool. And here in the NSGA we follow the method of assigning fitness values followed by the sharing fitness values.

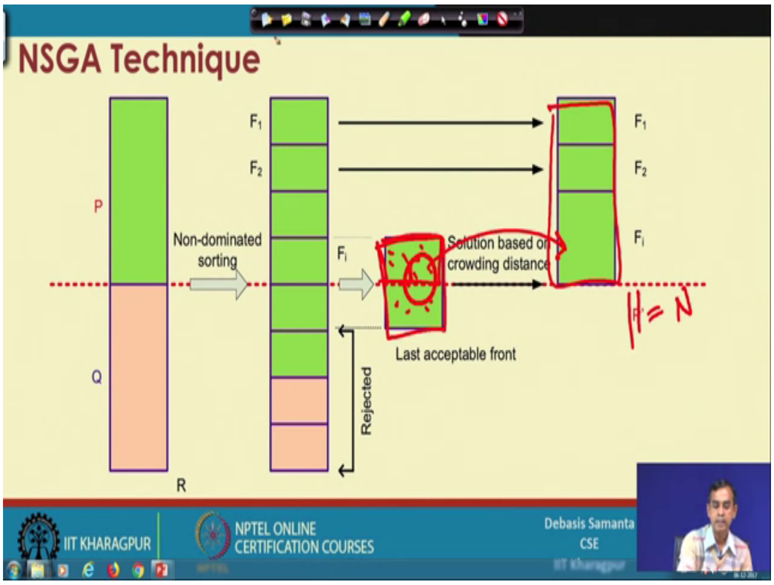
But in case of NSGA, the method that we follow for the selection is basically called the crowdring crowded tournament selection. So, in this lecture we shall learn about this crowded a tournament selection method so, crowded tournament selection method in NSGA 2.

(Refer Slide Time: 01:18)



Now as you know that crowded selection method is basically is crowded tournament selection method is basically required in order to decide from the last front to select the requisite number of solutions to fill this population of size F 1.

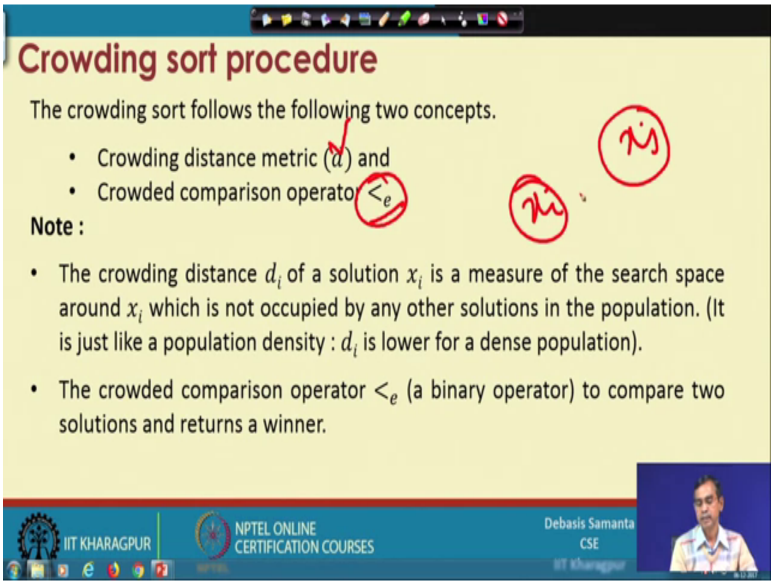
(Refer Slide Time: 01:44)



So, here if this is the last front that needs to be considered to fill this front, but if we include all the solution belong to this for it will exhaust the total capacity, from there we have to calculate only this amount of numbers to be included here so that the total size of this solution will be equals to N. So now, how to select the correct solutions are there the most preferred solution to this one, and this method is basically based on the crowded tournament selection method.

Now so, these are crowed tournament selection method will be discussed in these lectures.

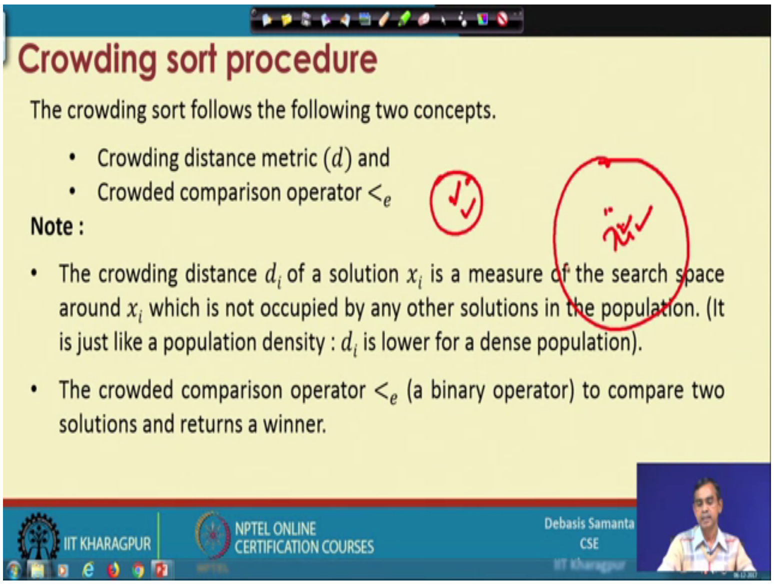
(Refer Slide Time: 02:27)



Now the crowded tournament selection method it is also called crowding sort procedure. It basically considered 2 concepts. First is that the measurement of crowding distance, and then another is called the crowding comparison operator. So, there is a one metric, if xi and xj are given, then how to find the crowding distance. Or rather we can say for every solution xi, how the crowding distance can be calculated. So, this is denoted as d, and then if xi, and xj are the solution, then how we can say that which is the winner that is based on the crowding distance measure that you have because, xi has it is own crowding distance xj is on crowding distance.

Then selection is based on an operator, that is called the crowding distance operator that can select the winner; that means, is compare based on this operator this one. Now crowding distance di if we denote of a solution xi is in fact, is a measure of the search space around xi which is not occupied by any other solution in the population. So, physical meaning of this crowding distances like this one.

(Refer Slide Time: 03:53)



If xi this is the one solution here, and then how many space it is there, by which the next the neighbor is there so this is the crowding distance. Now if this is the solution xi, and this is the crowding distance for the next neighbor it is there, then we can say that crowding distance of this solution is more than the crowding distance of this solution.

Now, another physical meaning is that if the crowding distance is very large. This means that this is a one solution, which is there in a less populated on region. On the other hand, if the crowding distance is small, then we can say that this solution is belongs to one solution, which is heavily the populated. So, is basically crowding distance says, that whether this solution is in a crowded region, or this solution is in a crowded region or both the solutions in a crowded region, then whose solution are in a heavily crowded region than the others.

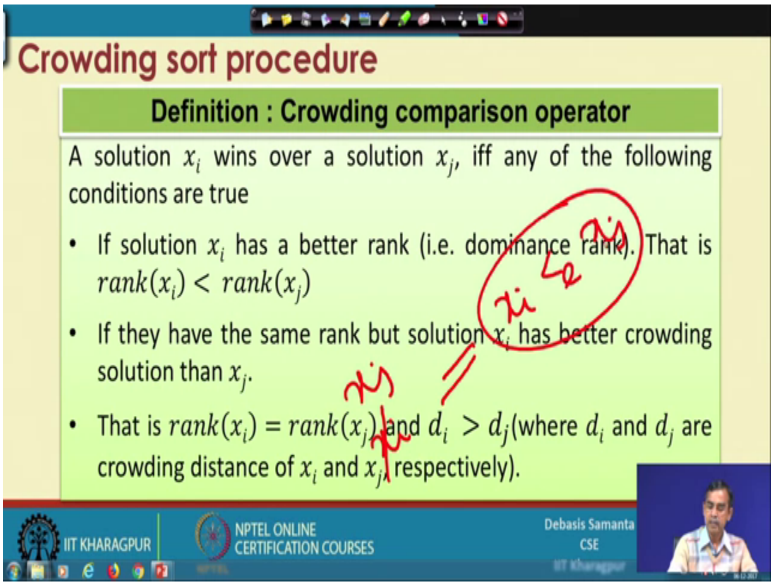
So, this is the meaning of this crowding distance concept it is there. And and then crowding operator this one is basically to compare the 2 solution, so far, their crowding distance is concerned. So, these are the 2 things are the here and we will discuss about these 2 things here. Now let us first define the crowding comparison operator that just now we have discussed by which the 2 solutions can be compared and based on this comparison we can select the best solution here.

Now, let us consider xi and xj are the 2 solutions, and they are the crowd crowding distance is known to us also. Now so, crowding comparison operator is basically is a operator which is defined here how this operator works for us. So, here the operator is defined like this, if solution there are 2 conditions actually, if solution xi has a better rank; that is, rank xi is better than rank xj; that means, xi in higher rank, I mean is a better rank than xj.

Now, rank actually you can remember I told once that all the some solutions which are the first part, they can be considered the higher rank, and then next solutions which are in the next rank is the next rank and so on. So, xi is the first front and xj is the next front then we can say that this one. So, here this is the one condition that is to be satisfied then you can say that xi is the winner than the xj. On the other hand, there is another condition if they had the same rank, but solution xi has better crowding solution then xj; that means, it has better crowding distance then xj then xi can be considered the winner than the xj.

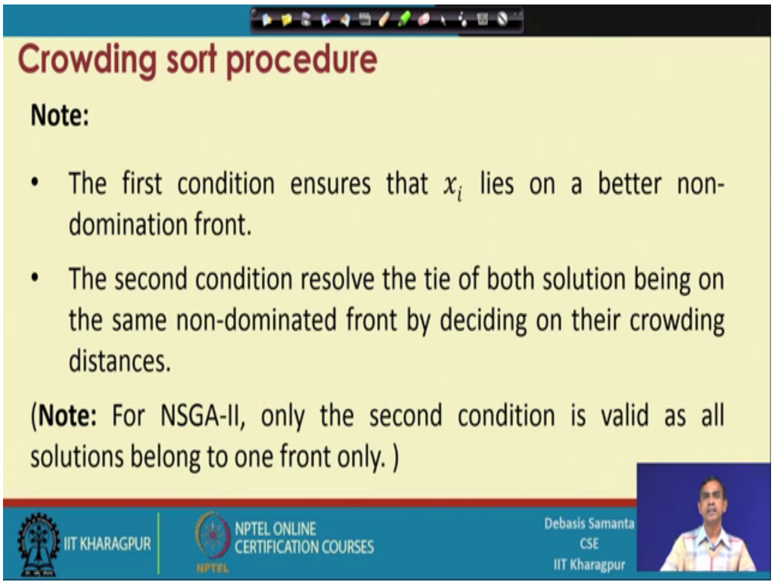
So, the 2 conditions are to be satisfied, and based on these things it will basically select the selects the winner. So, the conditions again rank xi if it is rank xj and di is greater than dj, then xi is the say operator. So, this is the idea about the crowding operator.

(Refer Slide Time: 07:19)



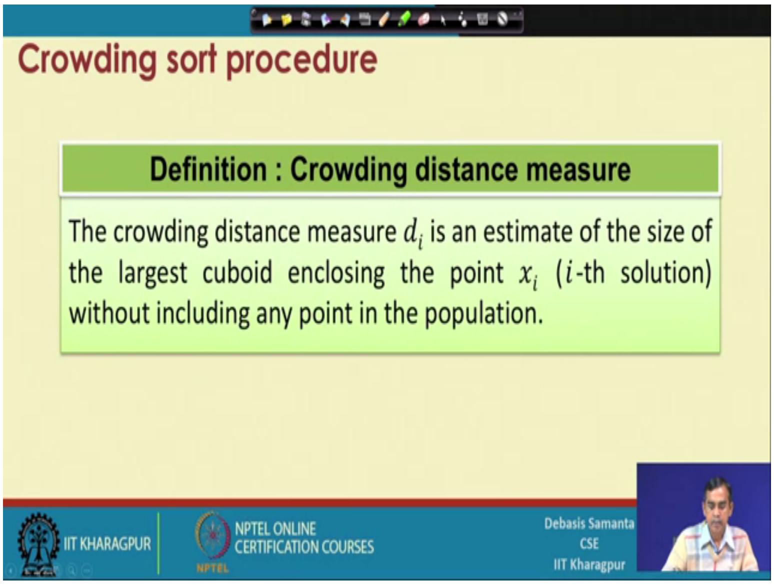
So, this means if xi, and this is the crowding operator xj is basically checked out of the xi and xj; which has to be returned. So, either xi or xj so, based on this condition, this operator is defined here. So, this is the concept that is there so far, the crowding sort is concerned now, let us see how this concept is there.

(Refer Slide Time: 07:45)



In our case, if we see in in the in our case; that means, we are to consider all the solutions belongs to a particular front that is the last front. So, therefore, rank is not required the first condition need not to be satisfied. Because all the solutions belong to this front have the same rank. So, that is why the first condition is not need to be checked there only the second condition needs to be checked. So, second condition resolved that i, when basically if both solution belong to the same front, but the solution that tie can be resolved by means of calculation of their crowding distance. Or in other words, in NSGA 2 only the second condition is valid as all solutions belong to only one front.

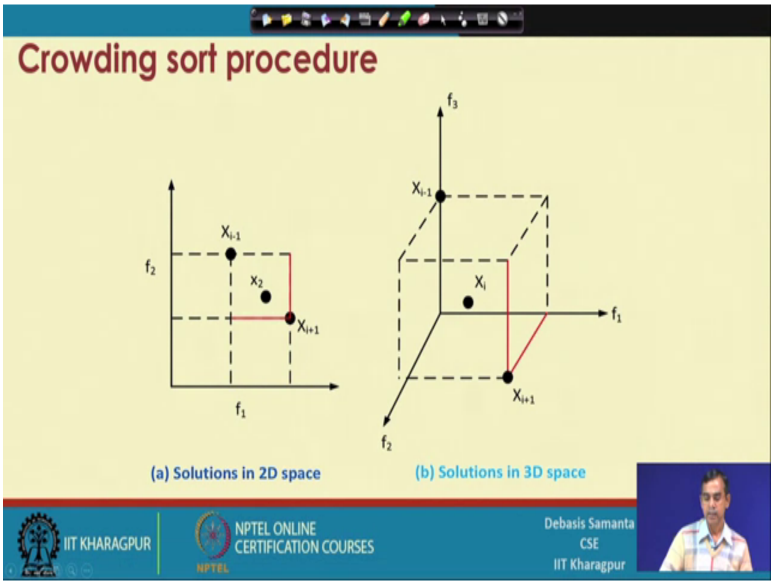
(Refer Slide Time: 08:42)



Now, let us see how the crowding distance measure can be carried out, we have already told you that crowding distance measure is basically the population density surrounding a solution. But how we can measure this population density? NSGA 2 follow a cleaver approach to do these things. So, according to this NSGA 2 crowding measure distance di for a solution xi; that means, is the ith solution is an estimate of the size of the largest cuboid enclosing the point xi, without including any other point in the population.

So, this is the definition actually. So, that is a largest cuboid enclosing the point xi, this is important ; that means, if xi is given to us, and if we are able to find the largest cuboid which surrounding the xi so that there is no any other point in that cube, then that cube will give a measure to the crowding distance. So, this is the idea about the definition of crowding distance and as it is there, I can illustrate the same concept with an example here, let us follow.

(Refer Slide Time: 09:56)



So, first consider this example, and it is a case of 2 dimensional; that means, 2 objective F 1 and F 2, and for solution say x 2 we want to find the crowding distance. Now surrounding this 2 point, the 2 nearest point with respect to x 2 is this one say xi minus 1 and xi plus 1. These are the 2 nearest point, with respect to the solution point x 2. Now then with x the 2 nearest points; that is, x 2 with respect to this we can find one what is called the region it is there.

So, this region is basically either crowding region here; that means, surrounding this x 2 these are the basically area by which no other points are enclosed. So, the x 2 is the crowding measure here, now the x 2 measures is basically the crowding measure is ok, we can take the calculation of the square of course, area, but this this energy to propose the major that this plus this is the measure of this crowding distance it is also alternatively, because if these 2 measures basically breadth and width measure in this case.

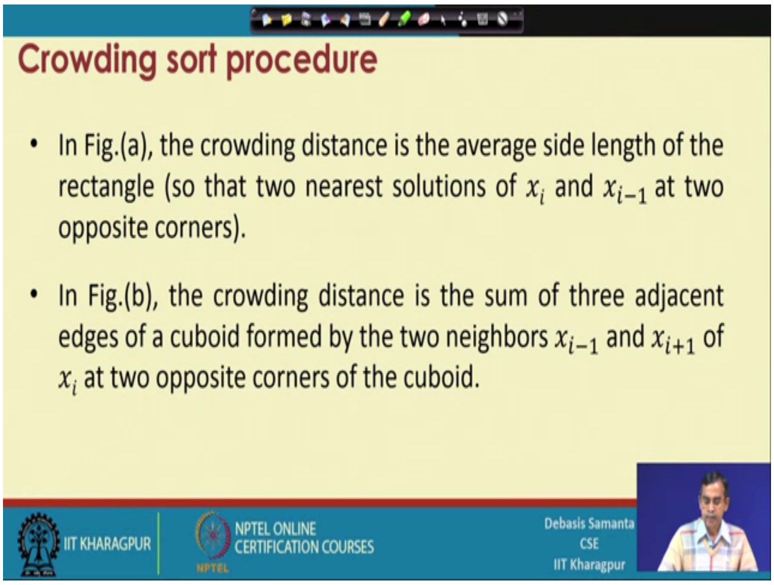
Now so, this way if we know x 2, and these are the 2 solutions are there we will be able to find these 2 distance and therefore, the crowding distance can be measured. So, this is the one example where the crowding distance how it can be measured here, and another example. So, this is another example here, 3 objectives are there. So, is a multi-objective optimization problem with 3 objective function F 1 F 2 and F 3? And we are interested to find the crowding distance for the solution xi. And suppose, xi plus 1 and xi minus 1 are the 2 solutions, which are the nearest to xi, they are the near most 2 solutions.

Now, if we can find the 2 solutions, which are near most to this one, then in 3 dimension unlike this 2 dimension we can find a cuboid. So, this is a cuboid and this one and then the crowding distance of this thing is basically this is the total area of the cuboid. But instead of calculation area of the cuboid, it will take the calculation of this and this are the measure of the size of the cuboid. So, this will give the measure of the cuboid, and then this can be given alternative measure or is basically the measure of the crowding distance.

So, we have learned about how the crowding distance for any solution in a 2-dimensional space can be calculated for any solution in 3-dimensional space can be calculated. Now extending the same idea, we will be able to calculate the crowding distance for any solutions, but in a N dimensional space. Now we will give an idea or the formula that that initiative others proposed it, how the crowding distance can be calculated will be discussed.

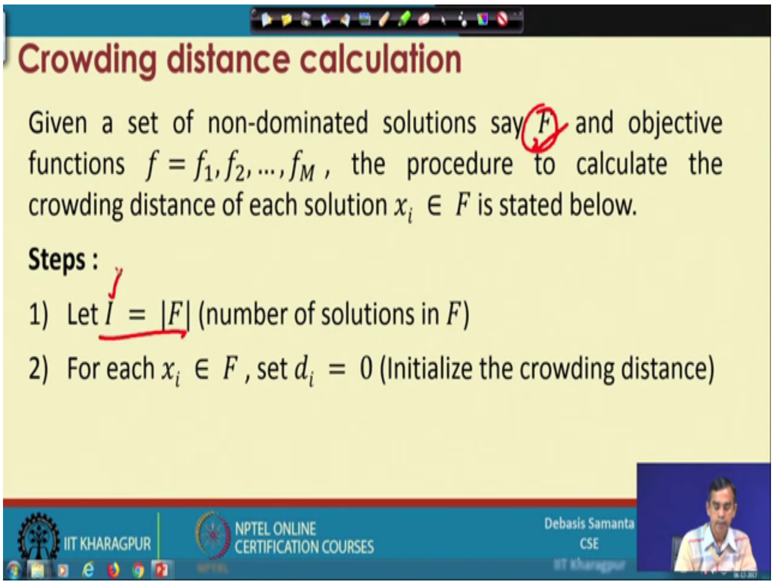
So, anyway so, crowding distance can be calculated, if a solution is given knowing the other solutions in the near dignity of the solutions.

(Refer Slide Time: 13:27)



Then the crowding so, is basically the crowding distance and then once the. So, the idea is that for all the solutions, which are belongs to a to the last non-dominated front for all the solution belong to the last dominated front. We have to calculate the crowding distance for all. So, this is the step that is required here.

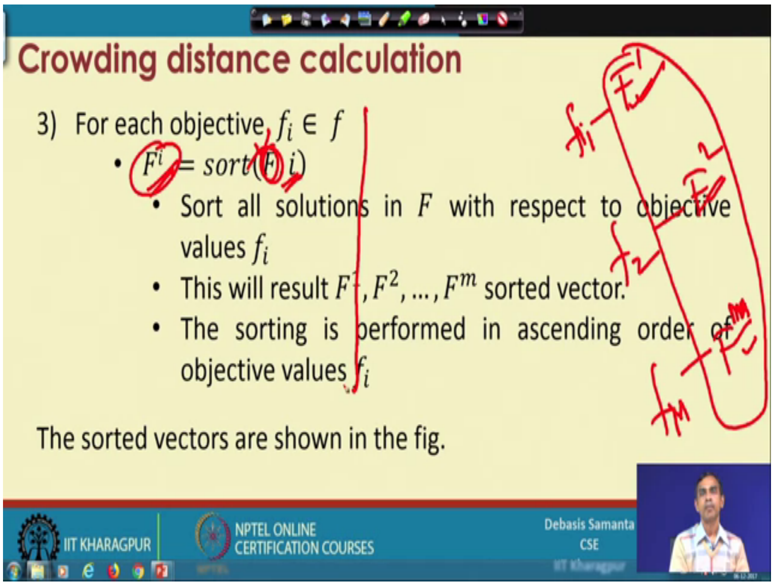
(Refer Slide Time: 13:57)



Now, crowding distance calculation can be carried out in a little bit mathematical way, that just I want to discuss it here. See suppose, given a set of a non-dominated front the last non-dominant front that is here let this be F. And they are our objective F objective function for each solution they are denoted as f 1 f 2 dot dot f M. So, for each solution we have this objective vector F with in terms of m objective functions. And let the size of the F be this one; that means, the number of solutions which belongs to the non-dominated front this be the i.

So, here basically the procedure is that for each xi in F set di equals to 0, initially the crowding distance of all the solutions is 0. And then we will calculate for each solution xi in F what is the crowding distance, we have to calculate it initialize 0, and then finally, you have to calculate the crowding distance for all solutions. Now here is the procedure for each objective fi in f, basically we will first sort all the set f, but with respect to the I th objective vector and this be the sorted f.

(Refer Slide Time: 15:09)

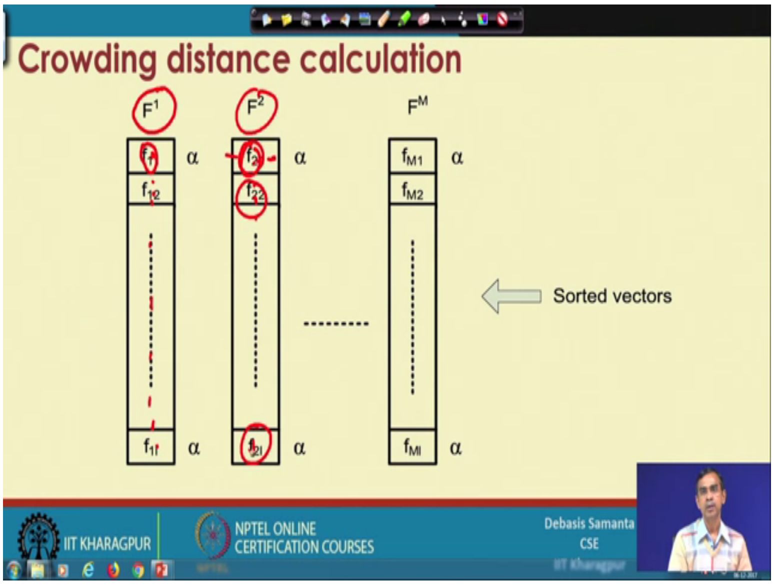


So, it is that idea it is like this.

So, first with respect to f 1, f 1 objective function we are to sort all the solutions which belongs to F and then the shortage solution will be termed as F 1. Similarly, with respect to f 2 if we sort all the solution belongs to F and it will give F 2. So, this way if with respect to f M, we can get the certain percent of sorry sort Fm. Now here we can see so these are the sets is a sorted order, but is a sorted order with respect to one component. This one sorted with respect to F 1, this one is sorted is F 2, and this one is sorted F 2, Fm and so on.

So, here basically a sorting techniques are to be followed by which all the solutions belongs to the set belongs to the set F can be sorted in terms of one objective function at a time. So, these are called the sorted vectors. So, pictorially all the sorted vectors can be shown like this, if you can see this figure here.

(Refer Slide Time: 16:46)



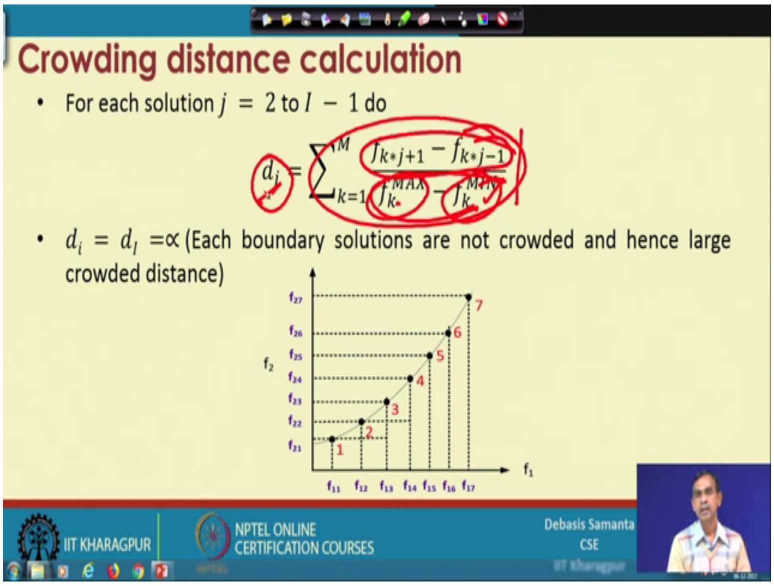
So, these are the sorted vectors, you can see F 1 sorted vector with respect to the objective function F 1. So, here basically all the solutions, they are sorted in ascending order, but with respect to F 1; that means, the solutions has the lowest value. So, for the F 1 is concern this is the next highest value, and it is the highest value of F 1 is concerned.

Now, again and this is the solution F 2 is respect to the second objective function; that means, with respective second objective function, all the solutions are there which has the lowest value of F 2, then this solution which has a next higher value and so on and this basically the solution which has the highest value; so, for the objective F 2 is concerned. So, this way the F 1 the sorted vector F 2 and F M, and here with respect to F M can be often.

Now, in this discussion we assume on concept is that here all the objective function are to be minimized. So now, if it is not minimized the other if it is to a maximize, then we will follow the descending order. So, if it is minimized, if it is maximized, then it is ascending order it is descending order. So, this procedure depending on, we can consider that all objective functions are to be minimized one. So, if it is a minimize, then all these are the ascending order of their objective function.

So, this way the sorted vector can be obtained. Once the sorted vector is obtained, then we shall be able to calculate the crowding distance between or crowding distance of each solutions easily. So, this method that is proposed in the NSGA 2 is like this.

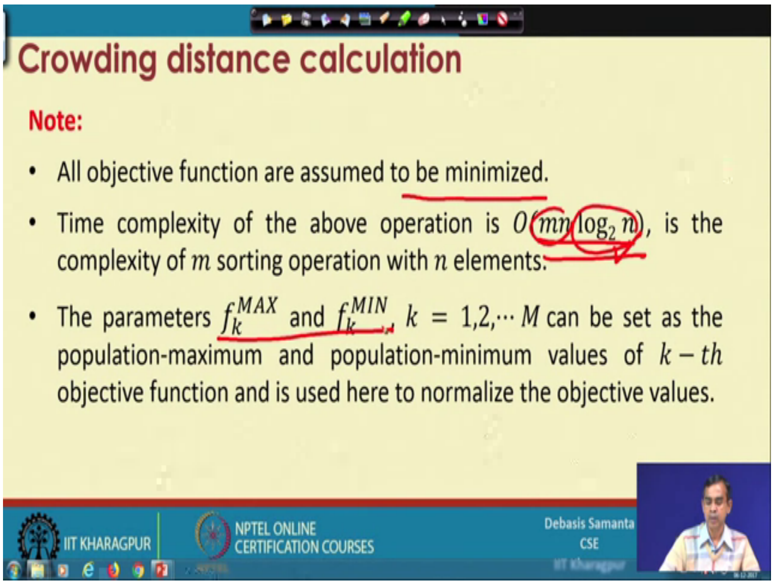
(Refer Slide Time: 18:35)



So, the crowding distance dj for any j th solution can be obtained like this one using this formula. This formula can be checked yourself, and you can find it here the fk max and fk min are the 2 values. It means with respect to the k th objective function or is the lower bound and is the upper bound. And this value is used to normalize all the solution, because is a normalization is require so that all values of the dj will be in a same range. So, this is for the normalization, and this formula can be verified yourself. And another thing is that, the first solution and the last solution this is basically di the last solution, because it is a boundary solution, they have crowding distance is infinite so, this is the condition.

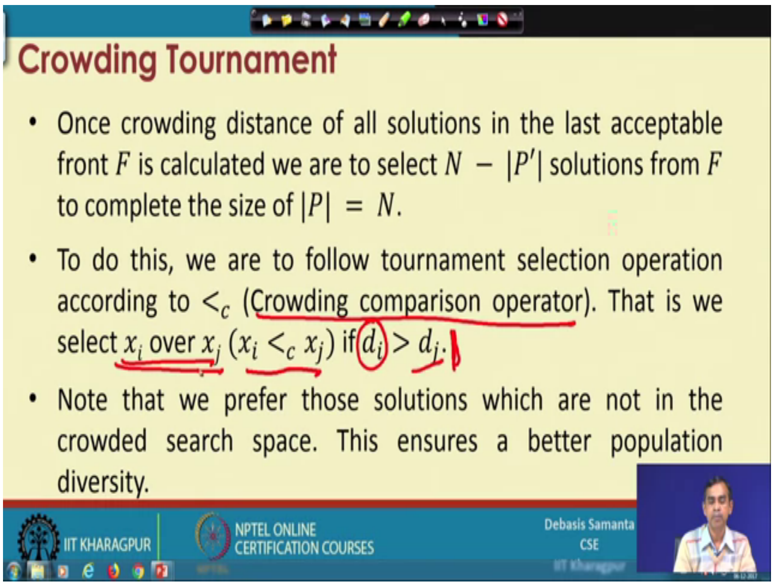
So, this way, we shall be able to calculate the crowding distance of all the solutions there. Now once the crowding distance of all solutions are calculated, then the then we can play the crowding sort procedure here ok.

(Refer Slide Time: 19:38)



So, here we have already you have mentioned that all objective functions are to be minimized in our previous discussion. And so, far the complexity of the crowding distance calculation is concerned, because it is a sorting method. So, it is a sorting complexity mn log 2 n where N is the size of the population. So, is the complexity is this one. And fk MAX and fk MIN are the 2 limiting values, and then we divide the this limiting a difference between the limiting values to normalize objective values ok.

(Refer Slide Time: 20:20)

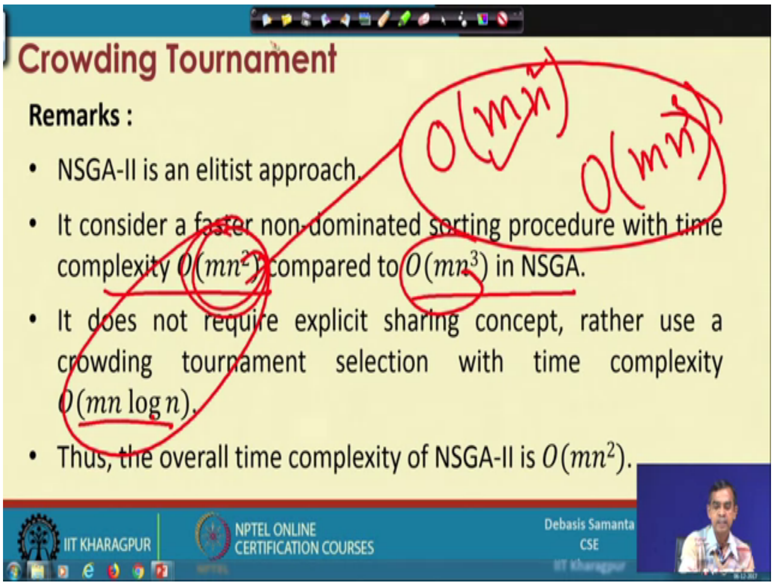


So, crowding distant calculation once it is done, then we will be able to play the crowding tournament selection game.

So, crowding distance selection game is basically it is like this we can follow the crowding comparison operator. Now here we ok. So, crowding operator can be applied to the 2 solutions xi and xj; that means, we have to see this one. Now here xi crowding operator xj, and xj xi will be termed as a winner if we see that, we are crowding distant di is greater than dj. So, in this case this is a crowding operator. Here and you can see again that all the solutions are get on the same rank, that is a you do not have to bother about rank.

So, here basically so, for the crowding distance based tournament is concerned, we prefer those solutions, which are not in the crowded search space. This ensures a better population diversity. So, basically di is greater than dj; that means, the solution xi is in not in a crowded region. Now so, this basically in to ensure the population diversity it is same concept that is where in NSGA, but NSGA follow. Then is count here instead of needs count it basically consider crowding distance. So, this is the difference between NSGA and NSGA 2 is here.

(Refer Slide Time: 21:48)



Now, NSGA 2 is an elitist approach. Why we said so? This is because, if you see the non-dominated front, it basically when we match the 2 solutions the previous generation and the next solution, and from there if we find the non-domination front, then it basically selects the all elite solutions first. So, that is why it is called the elitist approach. Because, first front, second front, third front and the elitist fronts are selected first and for the last front it is the lowest or what is elitist front from there we have to select using the crowding tournament selection.

So, here this is why the concept it is there. And so, far the procedure the time complicity is concerned. The total time complexity here is order of mn square compare to order of mn cube there in NSGA concept. And it does not require any explicit sharing concept that is therein can of in case of NSGA method. Rather, it uses a crowding tournament selections with and complexity order of mn log in. Now so, the 2 time complexity order of mn log in order of mn log in for the crowding tournaments and for this order of mn square for the non-dominated sorting procedure, putting together putting these 2 operations basically complexity the order of mn square only because this is the higher bound than this one. So, it is basically taken this one.

On the other hand, in case of non-NSGA it is order of N mn cube. And, obviously, So, it is a order of mn cube. And out of these 2 complexity, this is the time completely with the lower a port than this one. So, that is why NSGA is the first method. So, it is a first method, and it is an elitist method. So, this is the idea about the here, and accuracy it is observe that, this algorithm gives better result compared to both the compared to any pareto based approach that we know so far; that means, moga approach NPGA or NSGA it gives better result and with the fewer competition compare to NSGA of course. And obviously, if you consider time complexity, then it needs more time compared to moga and NPGA however, but the complicity is better.

So, these are the different pareto based approach we have learnt. And what I want to say in the summary is that, out of the different approaches to solve multi objective optimization problem, non pareto based approach needs a prior knowledge. Whereas, pareto based approach does not require any prior knowledge. So, this is the one advantage that the pareto based approach is there. And another difference between the non pareto and pareto that, non pareto gives only one solution. But all the pareto based approach gives pareto optimal solution; that means, tradeoff solutions. And then are from the tradeoff solutions, we have to decide one solution, and that solution require some posterior knowledge. That mean it is depends on your decision that out of these solutions which solution can be consider.

But in case of pareto based solutions or pareto optimal solutions, we can select any solutions out of a large set of solutions that can be said that can satisfy your requirement. So, this is the difference between the pareto based and non pareto based approach. And as we told you the non pareto based approach it is applicable, if we see that only few are tradeoff solutions are to be considered or it is there in the problem solving.

On the other hand, we should apply pareto based if we see that there is a large number of solutions are possible which are equally towards the optimum solution. So, in that case we follow pareto based approach. Out of this non pareto and pareto based approach in fact, people prefers pareto based approach because of it is accuracy and then performance first. So, this is the technique that we have learned about multi objective optimization problem solving using genetic algorithm.

So, our next topic in the next field will discuss about a neural network concept to solve some computing problem in different application.

Thank you.