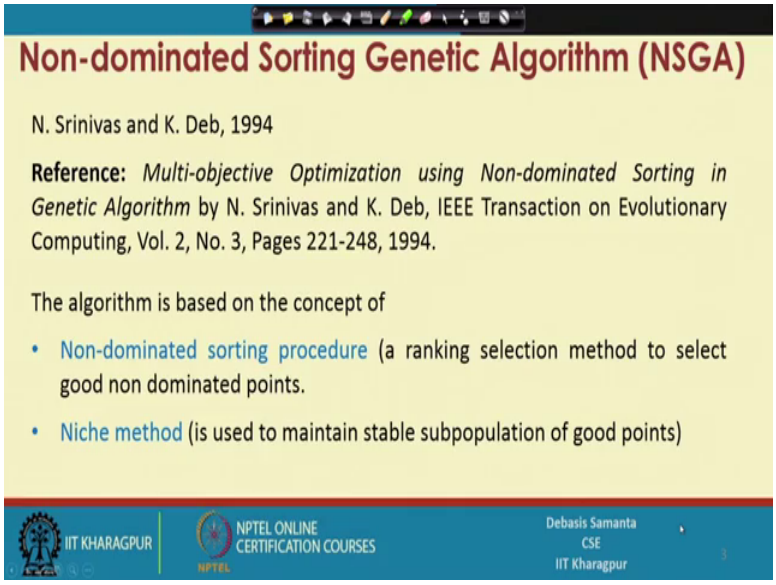
Introduction to Soft Computing
Prof. Debasis Samanta
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 31
Pareto- based approaches to solve MOOPs

So, we are learning about different approaches to solve multi objective optimization problem, in the last few lectures we have learned different approaches belong to non pareto based and pareto based approach. We will continue our discussion today we will learn few more pareto based approaches. So, first we learn about the most popular 1 compared to the other pareto based approach it is called non dominated sorting genetic algorithm shortly it is termed as NSGA.

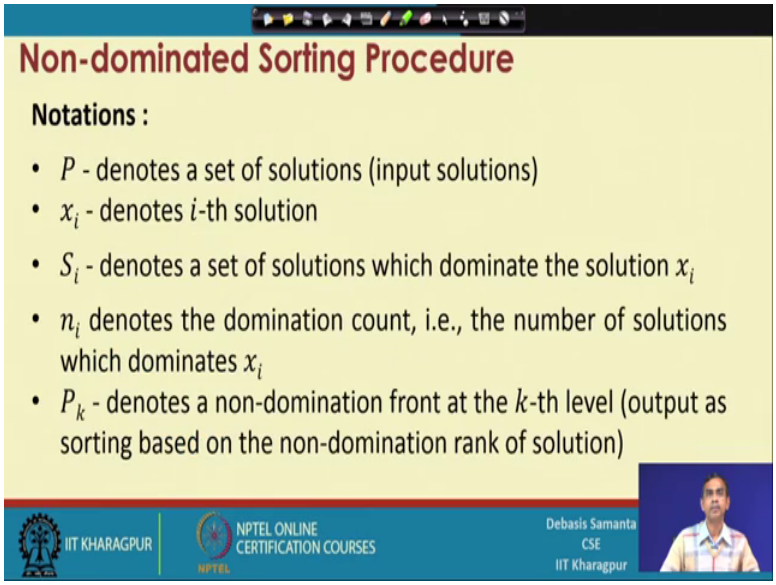
(Refer Slide Time: 01:00)



So, today learn about this the technique the NSGA algorithm, this algorithm first time proposed by N Shrinivas and K Deb in 1994, they have published this work in the journal called IEEE transactions on evolutionary computing, and the title of the paper was multi objective optimization using non dominated sorting in genetic algorithm. Now this algorithm is different from the previous algorithm in the con cons in the in the in terms of the concept that they followed. So, it is basically based on the concepts of non dominated sorting procedure, and then they used another method another concept it is called the Niched method.

So, they basically select the non dominated front by means of non dominated sorting techniques, it is basically a ranking based selection method to select the good solutions which are to be in the non dominated fonts, and then in order to have a good population diversity, and then better selection pressure they follow while they assign the fitness value to the solutions, and they follow one concept it is called the niched sharing. So, we will discuss about the 2 concepts, and then we will know exactly how NSGA the algorithm works.

(Refer Slide Time: 02:43)

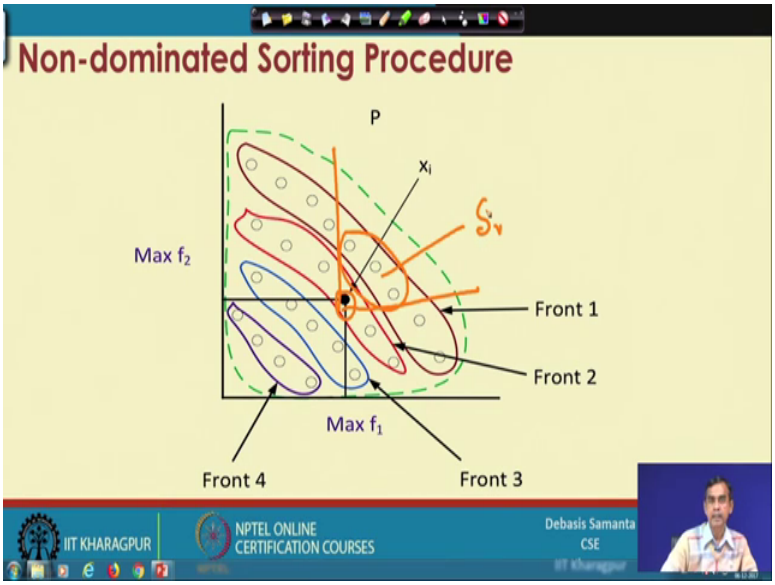


Now, in order to understand the algorithm, we will consider few terminologies. So, suppose P P denotes P denotes a set of input solutions basically given the input solution. It is basically the current population we can say. So, we have to find for the current population all the solutions which are the, which are lying on the non dominated front. So, P is the current solution P is the current solution. So, P denotes the current solution and x i x i denotes any i-th solution in the current set. So, x i is a solution which belongs to the set P, we denote S i S i is a set of solutions it is a basically sub set of P which basically contains all the solutions which dominate the solution x i.

So, S i contains all the solution which dominates x i, and then n i denotes the domination count it basically defines the number of solutions which dominates x i, and we also denotes another notation called the P k denotes a non domination front at the k-th level

anyway. So, we can explain all this terminology in with an example so, that we can understand about the concept.

(Refer Slide Time: 04:37)

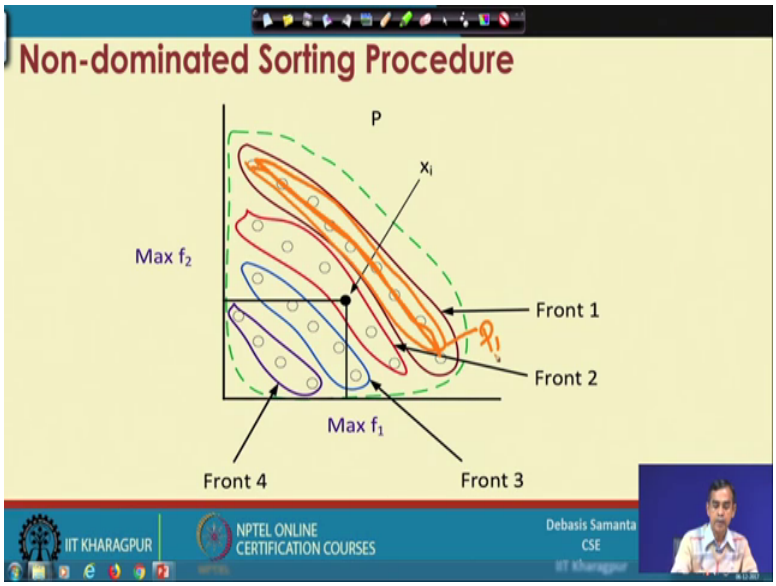


Let us follow this diagram in this diagram. So, this basically denotes all the solutions, we say this is the P, P is the set of all solutions, and then we consider any solution x i. So, if x i and in this case we consider that f one is the function to be.

Maximized and f 2 function to be maximized both the function to be maximized if. So, then this x x i is the solution, and then in this case the S i who is denotes the solution who dominate x i. So, in this case these are the solutions. So, these are the basically solution if this is this solution, then these are the solution basically called S i these are the solutions which dominates this one, and then we defined the domination count n i. So, basically by which all the solutions is basically this solution dominates all other solution in this region.

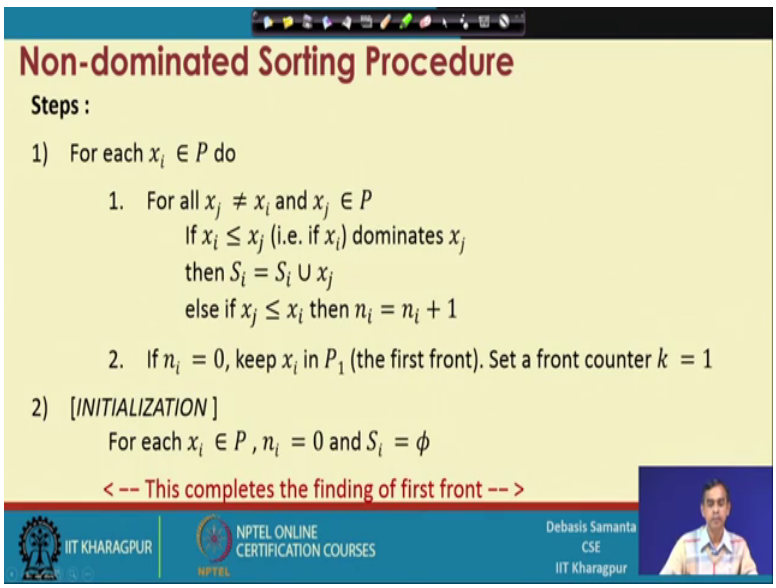
So, so the number of solution in this is basically n i the solution count of x i, and the front. So, this is the solution which is basically the non dominated solution because there is no solution which is dominated by this. So, these are the solutions is a non dominated solution, and it basically creates a front we can say this front is 1 1.

(Refer Slide Time: 06:16)



So, it is basically this is the front P f 1, the first front now if we remove all the solution from this front, then it will give another front, the next front if we remove all this solutions from here, then the next front is basically next non dominated front or next front. And similarly if we remove all this solution, then next front and next front so, here the different front so front 1, front 2, front 3, front 4 like this one. So, these are the concept that is there. So, we have learned about the different terminologies. Now let us see how the algorithm can be defined here, and what is the procedure there in the algorithm.

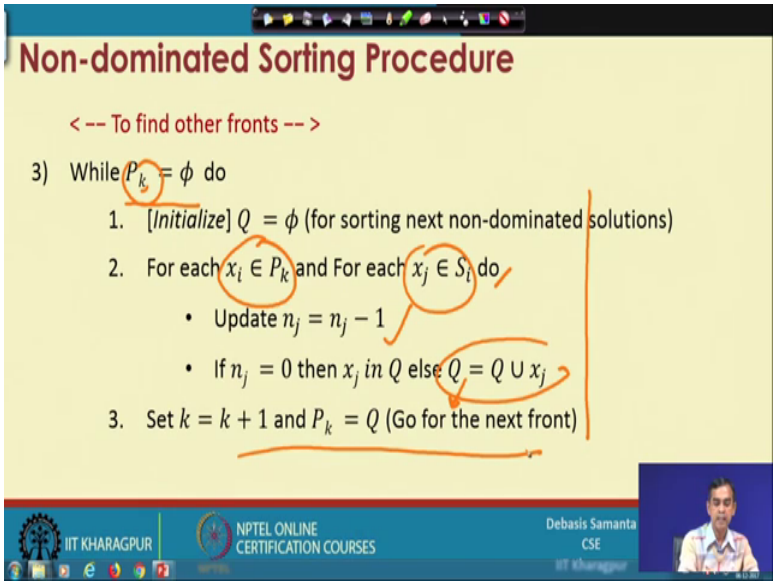
(Refer Slide Time: 06:55)



Now so, the algorithm basically find the non dominated front, and then the other dominated other fronts. So, let us see how this algorithm works. So, here this is the code that we have given it here this code is basically to find the non dominated front that dominates out of most front, and here basically for each solution x i which is in the set P, and for all other solution x j such that it is not equals to the x i and this x j belongs to P then we will consider this one, if x i is x i dominates x j, then we put x j into the set S i. So, S i is basically the solution which basically dominated by x. So, it is basically if x i dominates x j when it is input in the solution S i and if it is not dominated.

So, it is there; that means, it dominates x j dominates x i, then you can count the domination count. So, n i equals to n i plus 1. So, this way it basically count the solution domination count as well as the solutions the dominated sets, and ; obviously, if n i equals to 0 for all the fronts that is there, then we can say this solution S i is basically the P 1 that is mean the first front, and you can say k equals to 1. So, this basically the front and it is called the first front or non dominated sorting front the first non dominated sorting front.

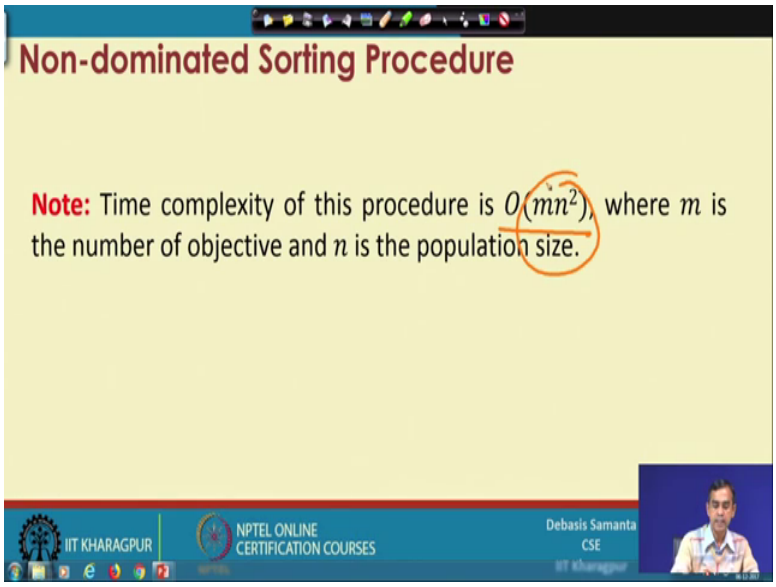
(Refer Slide Time: 08:44)



And then we can find in the same way the other fronts. So, here the procedure so, it is basically initially say the k-th front, So, P k initially it is phi because initially there is no element in the front then you if repeat this steps, then it will create the front P k. So, for each solution x i in P k P k is basically the first it is it is start from the first front P 1, and

for each solution x j in S i we have to do this one. And then we can add into the Q. So, finally Q will give the next front after the P k front. So, this way we can find all the solutions all the other fronts in the solid state of solutions.

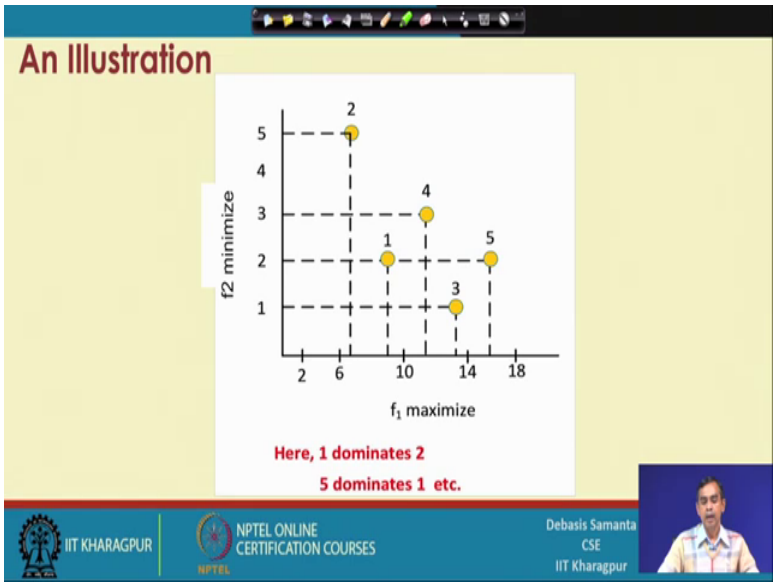
(Refer Slide Time: 09:35)



So, this way we can find is a little bit of programming approach, we can find using the concept of domination the first front, second front, and then n-th front depending on the number of solutions are there. Now so this programs I have mentioned this program in order to understand that how much time that it will take. So, it can be observed that the time complexity of this procedure of finding non dominated sorting fronts, or it is basically called a non dominated sorting procedure.

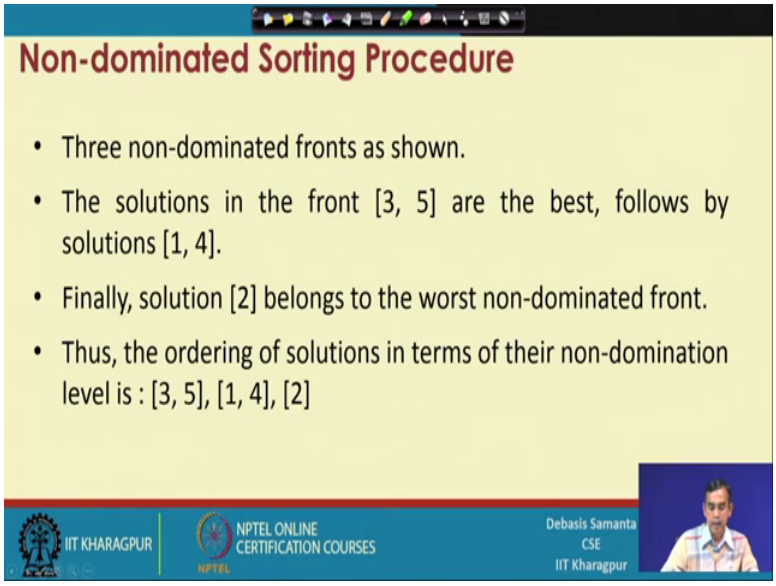
The time complexity of this procedure can be seen it order of m into n square, where m is the number of objectives, and n is the size of the population. So, it is basically in terms of this complexity, and regarding this complexity, we will discuss few more things later on. So, this is the procedure the non dominated sorting procedure is the main one critical procedure it is there in the in the NSGA algorithm.

(Refer Slide Time: 10:33)



Now, again I can illustrate this concept let us consider these are the solution set here, these are the solution sets here, and here f 1 is maximize and f 2 is minimize. So, if it is like this then we can easily understand that. So, this is these are the solution which are basically the first front, we can easily identify, and then this is the solution which is the second front, and this is the solution using the third front. So, in this concept there are 3 fronts, as we can say and the 3 fronts are like this 3 fronts are like this.

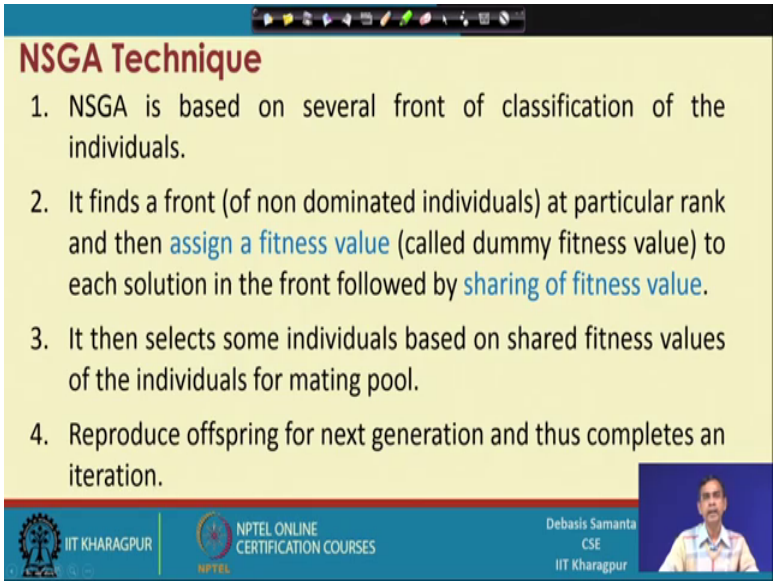
(Refer Slide Time: 11:23)



And so, here you can say that the solution in the front 3 5 are the best solution, and they are basically called the non dominated solution, on the other hand this solution 1 by 4 the next inferior solution, and the finally the solution 2 are the worst solution. So, for the front is concerned so these are the different fronts, and the different solutions are there. Now if we order all the solution based on their this front, then we can say these are the ordering.

So this is the first front second front, next front, and so on so on. So, this ordering is called non dominated so ordering right. So, what I want to emphasize is that a given a set of solution, we shall be able to find using the non dominated sorting procedure, the different fronts and ordering of the different fronts, and so for the ordering is concerned. We can say all the solutions which belong to this front can be termed with rank 1, and this is the solution rank 2, and rank 3. So, non dominated sorting procedure a allow us to find the rank of all the solutions given a set of solution or current solution set. So, this is the one important concept that that is fundamental, there in case of non dominated sorting genetic algorithm.

(Refer Slide Time: 12:57)



And then we will discuss about the basic techniques which is followed there in NSGA NSGA algorithm. So, first we have already mentioned that it basically based on the classification of several front that can be there in the set of solution, and it finds in fact, it finds a front at particular rank, and then basically assign a fitness value.

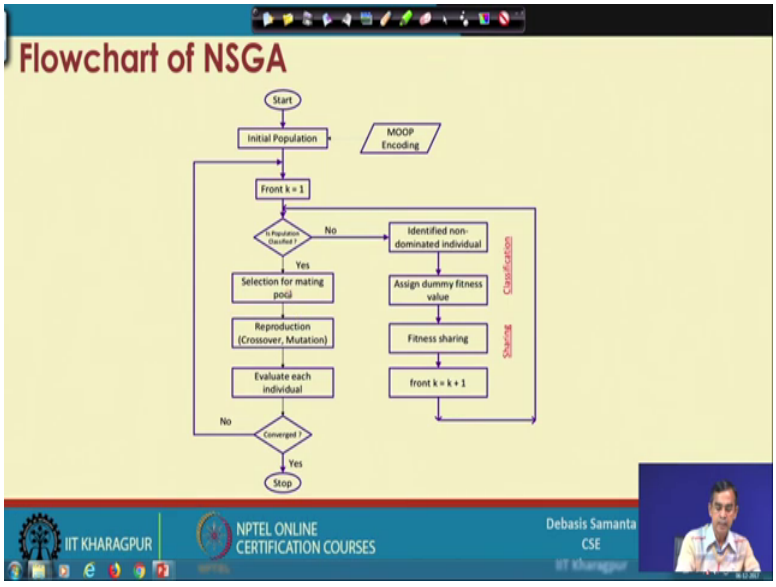
So, we will discuss about these are the concept that ok, we know exactly how to decide a rank of the solution this basically deciding the rank of the solution by virtue of calculate following the non dominated sorting techniques, once the rank is known to us then we shall be able to assign a fitness value. So, this is one important task that is followed there, this fitness value is also called dummy fitness value to each solution in the front, and then the next technique that it follows is basically sharing of fitness values.

So, now our task is to understand how to assign a fitness value that is the dummy fitness value to each solution, and then how to share the fitness value, and what is the rational behind this things also we shall be able to learn. So, basically once this fitness value is assigned and sharing a fitness value, this means that each solution will be given a fitness value as for the NSGA technique. Once the fitness values is assigned to each solution then it basically consider the solution based on their fitness value to be considering the mating pale, mating pool, and then they will be considered for the reproduction here. The reproduction procedure is same as the genetic algorithm.

So, basic idea is that we have to create the mating pool, and for this mating pool creation this is the selection strategy that it follows, and selection strategy means it basically find the non dominated front, and then followed by that assigning the dummy fitness value.

And then finally the sharing fitness value so, this basically gives the fitness calculation to all the solutions, and based on the fitness calculation we shall follow the reproduction.

(Refer Slide Time: 15:15)

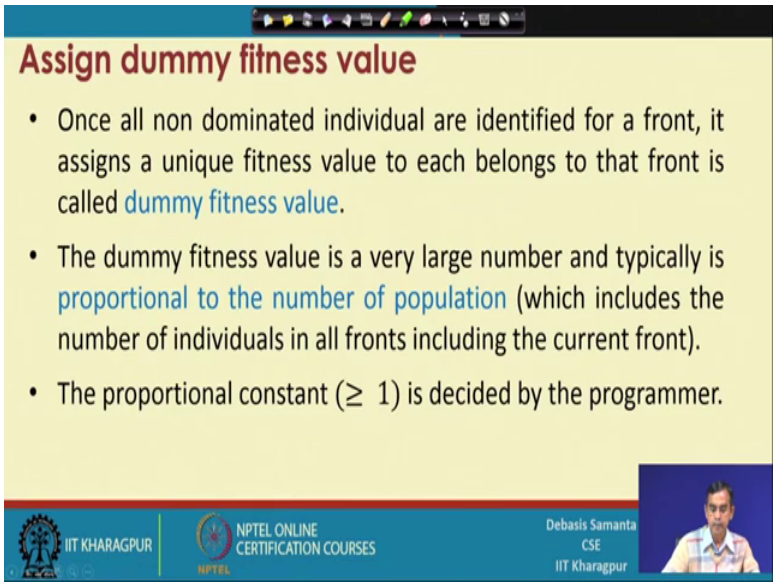


Now, so this diagram shows the flow chart of the algorithm, now if we little bit carefully check it. So, it is basically start with the initial population, and then it will basically consider the classification of the front, and then sharing of the fitness value assigning the dummy value, this is the step which is there. Now here basically we start with the first front k equals to 1, and then identify so k equals to 1 basically we consider the first front, then because the k-th front and then assign dummy fitness values and dummy fitness sharing. So, this basically for the k-th fronts. So, k starting with 1 and this will procedure.

For all the solution until all the solutions are assigned the fitness value and sharing this one. Once all the solutions are assigned some fitness value, then it will come here it is basically selection of mating pool, the selection can be the same selection that is there in g a technique like roulette wheel selection. Once the mating pool is selected then the reproduction procedure, and then evaluation of the solutions, and if we check the convergence criteria reached or not if reached this is the solution that can be obtained here, otherwise it will repeat the procedure for the next generation. So, the idea it is like this and so, basically if we see this is the conventional g a framework whereas, these basically the idea about how the selection techniques is different than the conventional g a technique.

So, this is the flow chart of the algorithm NSGA. So, there are mainly 2 tasks classification and sharing, and then we will discuss about the sharing concept how the sharing is there, and then sharing is basically use the fit dummy fitness value assignment. Now let us see how the sharing and dummy fitness value assignment is there in the NSGA 2.

(Refer Slide Time: 17:14)



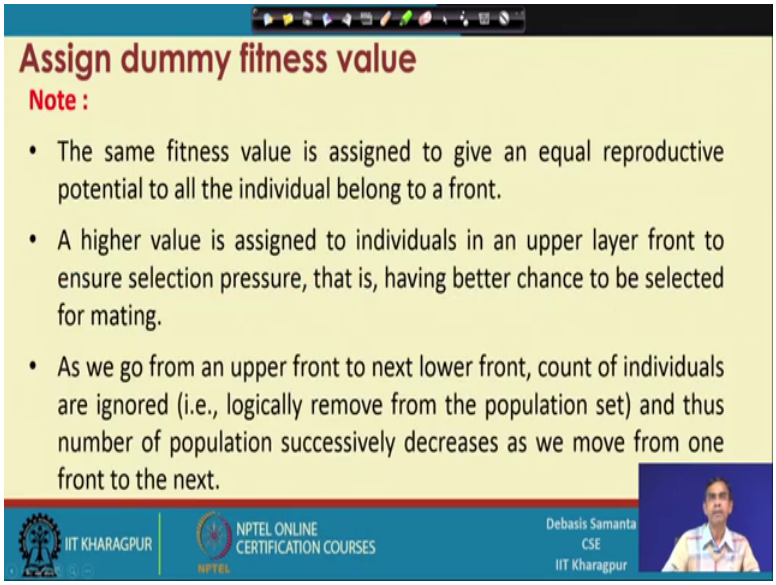
So, here the basic idea about assigning dummy fitness values is basically we first find all the fronts, then assign dummy fitness value to each solution belongs to a particular front. So, here basically the idea it is that dummy fitness value is a very large number, and typically it is proportional to the number of population, which includes the number of individual in all front including the current front. Now here is the idea it is that at an instant if this is the one solution.

We have to assign a dummy fitness value it basically assign 1 number which is very large number the number is denoted by all the solution which is in other which is not in the which is in this front as well as all the fronts below this front. So, it is the concept all the solutions, and then plus all the solution in this for include this is the large number of values are there and this assign.

So, it is again you can see all the solution which is there in the front is basically assigned the same dummy fitness value. So, the this way we can assign the dummy fitness value, and it is a proportional number sometime proportional constant will be greater that 1 to have a very good number. So, this basically assign the dummy fitness values to all the solution, and you can say that if we assign the dummy fitness value to the non dominated front; that means, the first front they have the very higher number than the next front and so on.

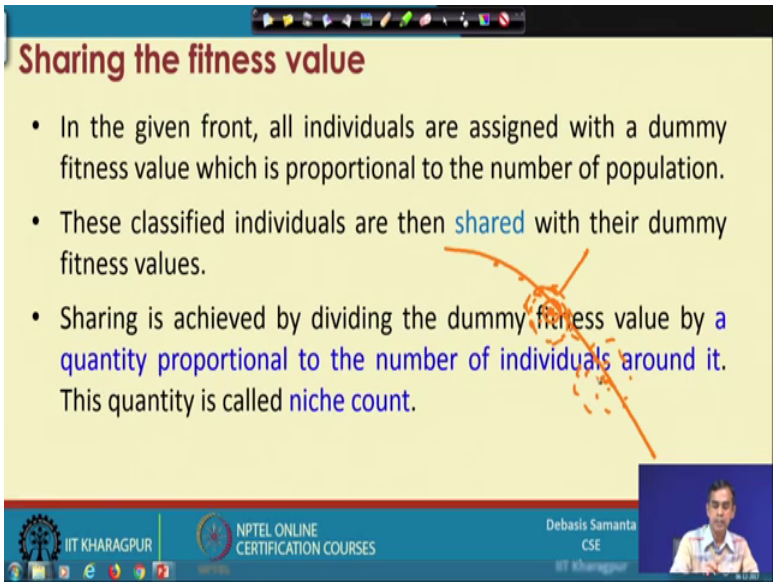
So, on as the front goes higher the dominant fitness value assign goes lower. So, this way we can basically give the fitness values is a (Refer Time: 19:00) fitness values now. So, that can be very large and for the best solution for the superior solution, has the large number of what is called the dummy fitness value is compared to the inferior solution to that.

(Refer Slide Time: 19:19)



So, this basically assign assigning the dummy fitness values. So, here actually we can summarize the things that is there, the same fitness value is assigned to give an equal reproductive potential to all the individuals belong to a front. A higher value is assigned to individuals in a upper layer front to ensure selection pressure that is having better chance to be selected for mating, and as you go from an upper front to the next lower front count of individuals are ignored that is logically removed from the population set, and thus number of population successively decreases as we move from one front to the next front ok. So, this is the idea about dummy fitness value assignment.

(Refer Slide Time: 19:59)



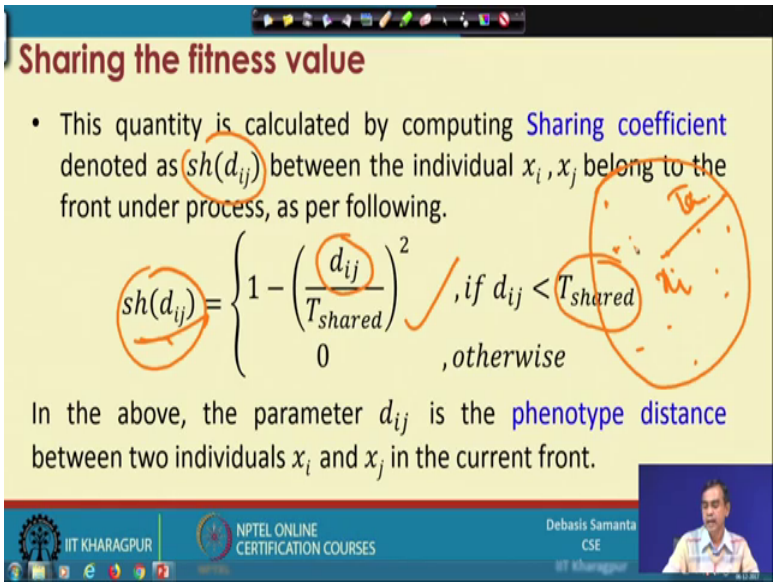
And our next concept there in the NSGA sharing the fitness value so, here basically we have understood that assigning dummy fitness value, the main purpose is to have a very good selection pressure, and then the sharing the fitness value the objective is to that how the how population diversity can be there. So, that we can find better solution and the solution cannot be tapped into the local optima. .

So, here the idea about sharing the fitness value is like this. So, in the in any front all individuals are assigned with a dummy fitness value which is proportional number of population. This individuals are then shared with their dummy fitness values. So, it is called the sharing concept, now sharing is divided a sharing can be of sharing is achieved by dividing the dummy fitness value by a quantity proportional to the number of individuals around it.

Now this is the one concept it is basically the niche concept. So, it is niche count. So, here the idea is that if so this is the one solution, and it 1 front, and we have already assigned 1 dummy fitness value. So, we have to see right we have to share the fitness value in the sense that, now we have to see what are the solutions which is around this one right. So, if this is the solution which has had a less solution around this one, then it has less niche count then this one. So, dummy fitness value that been assigned to this solution, if we divided by this assigned to this solution, if we divided by this niche count, then it is called the shared fitness value. So, the solutions which has very large niche

count has the higher fitness value, and compared to the solution which has less niche count has the less shared fitness value. So, this way all the solutions when assigned the dummy fitness value, when a particular front have the same dummy fitness values after the sharing information they even all the solutions are same front have the different fitness values, and that is the ultimate fitness value that should be considered in the mating pool selection procedure. So, this is the idea about the concept niche count.

(Refer Slide Time: 22:15)

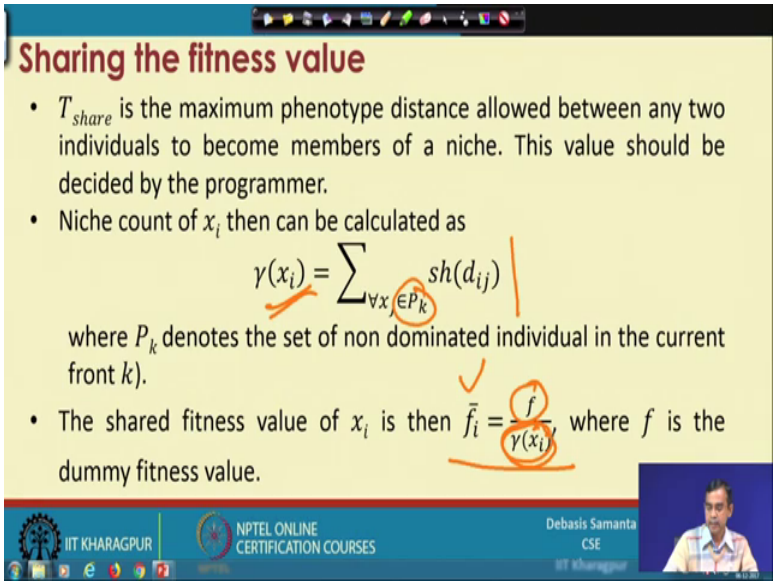


Now, so niche count how this niche count can be consider can be calculated, the idea it is the same idea once we have discussed about sharing the niche values in the NPGA algorithm, it is the same concept it is followed there in NSGA. So, here basically we say d i j is basically the distance between 2 solution x i and x j, and again another constant that needs to be decided it is called the T shared. So, it is basically if this is the solution x i, and T shared is denoted how far we have to consider that mean what will be the region of sharing basically. So, it is x i and this is the radius it is called the T shared.

Then whichever the solution which is there it will be considered the shared d j count sh d i j, and it is the formula this one. So, this is the same formula we have used there in NTGA algorithm. So, finally, for this solution and then for all other solution within this region it will calculate sh d i j, and then S i sh d i j is basically the sharing information or it is basically the niche count. So, these are the niche count.

Now sh d i j it is higher, if we find the number of solutions higher, and it is lower if the number of solutions this one. So, it is basically niche count; that means, it is basically how a solution is in a region which is basically having the higher population than city compared to the others. So, the solution which has the higher population density has the higher value of niche count compared to the solution with the less population density. Now so, here basically the concept of how to niche count for each solutions that can be obtained once this solution is there, then we will be able to consider about ok.

(Refer Slide Time: 24:09)

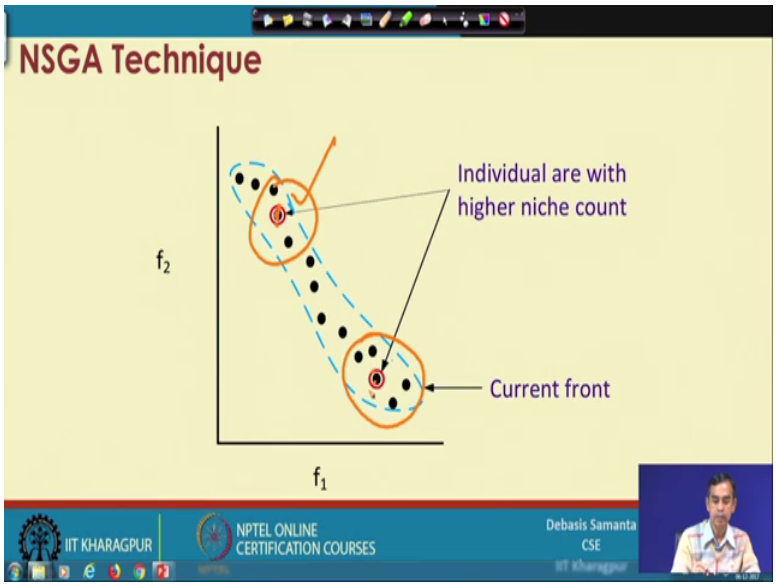


So, this basically the total niche count, if you consider all solutions with respect to the i-th solution belongs to particular front P k, and then we can share the fitness value if f is the original fitness value of the i-th solution, and divided by the gamma x i that is the niche count of the solution it is there. Now we can understand that the solution which has the lower population density has the fitness value it is this one higher fitness value, and the which has the niche count higher it has the lower value. Now it is little bit confusing like, actually it is the idea is that sometimes the good solutions should be combined with the bad solutions, this because population diversity is much more. So, if we consider all good solution then population diversity is less.

So, we have to share the fitness value. So, that the fitness value can be can be shared among the little bit solutions which are not so, good solution or inferior. So, share count is basically say that how a solution can be a good representative than the other solutions

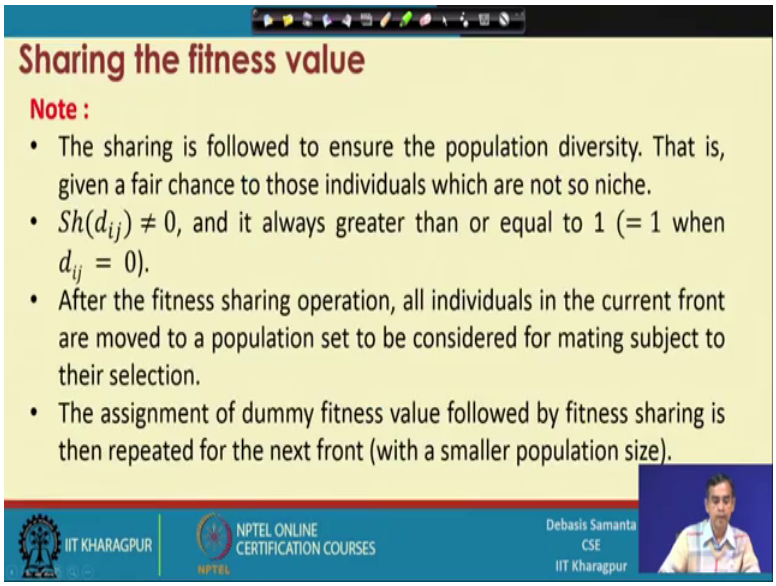
are there. So, this way we will be able to kind that this one. So, this basically the shared fitness value and these are final values that needs to be considered for all solutions before the mating pool creation.

(Refer Slide Time: 25:41)



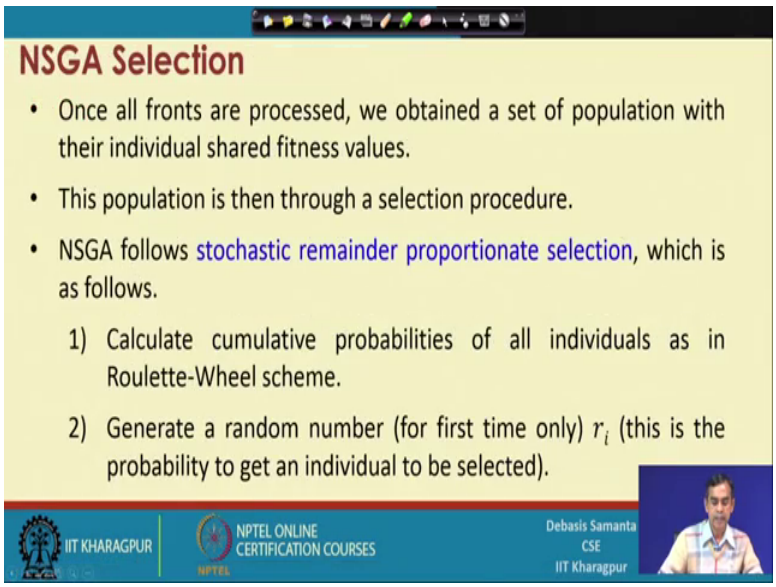
Now, so the idea it is like this so we have discussed about. So, this solution and this solution if this is this one, it has the niche count is much more then this solution has had the niche countless. So, we will select we will prefer for the mating pool compared to if their competitor, then we will prefer this solution than this one. All though all solution I have the same dummy fitness value, but whenever divided by this shared niche count, it is it gets more weightage than this one. So, this way we will be able to have maintain the population diversity.

(Refer Slide Time: 26:18)



So, this way you can learn about how the solutions the particularly belongs to a particular front can be shared, and then we can consider for mating pole creation.

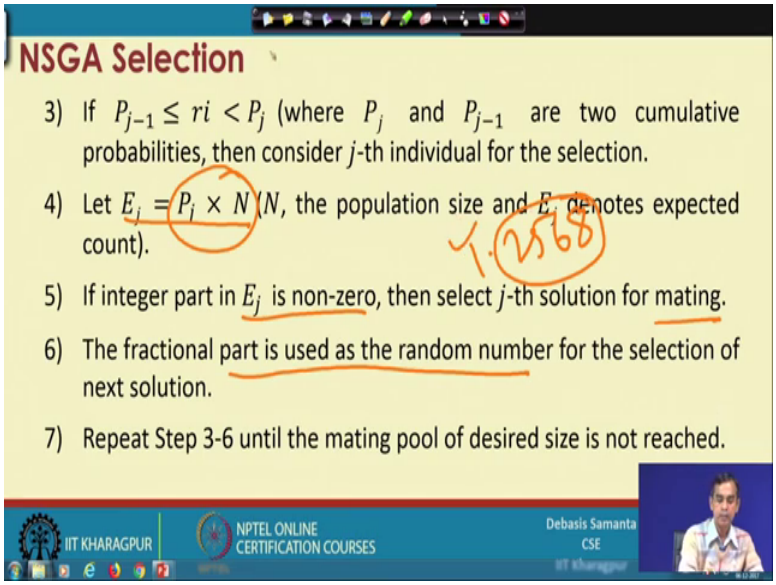
(Refer Slide Time: 26:31)



Now, so once it is there then we have to go for the selection procedure for the mating pool creation. So, all the solutions on the particular front, we have to follow one method called the stochastic remainder proportional selection technique which is discussed here, it is basically a (Refer Time: 26:48) roulette wheel selection technique, and generate

basically random number r i, and then follow the random number to consider the same procedure a roulette wheel in a cumulative probability is concept.

(Refer Slide Time: 27:03)



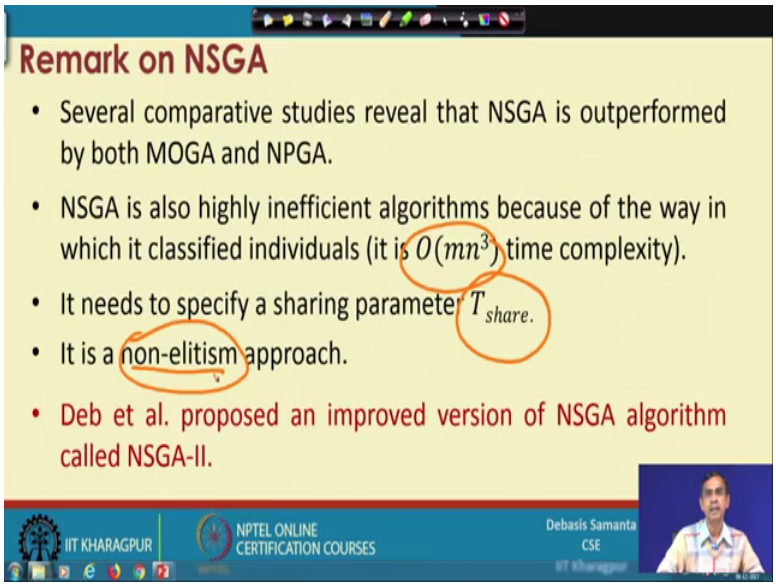
That can be consider and it can be so, used to select the solution there. So, this basically the same procedure that is there in the roulette wheel selection and, but here the one difference is that in this method is basically it calculates based on this r i, and selection of the particular j-th solution, it calculates the expected count, we have learned about expected count while we have discussing about the roulette wheel selection method.

And then it basically the ok, in the previous roulette wheel method that we have learned, then we have to generate the random number each time, but it does not require to random generate number, in this procedure it basically follow the calculation of E j which is using this formula, and then if E j is non 0, then the solution is selected, and then for remaining; that means, for the a non integer parts we can use non 1.2568 so is basically this is the next random number to be considered if it is there. So, this is the procedure it is followed instead of generating random number it will follow, this concept so that we can avoid the compression (Refer Time: 28:09) there.

(Refer Slide Time: 28:13)

So, this way the selection can be done, and then mating can be for sure, and the reproduction procedure same as the reproduction procedure there in the simple genetic algorithm.

(Refer Slide Time: 28:22)



Now, I just want to conclude before going to stop, this discussion here, the so it has been observed that if we compare the NSGA technique with respect to MOGA, and NPGA it is observed that in most of the cases NSGA gives better result compared to the MOGA and NPGA; however, the main drawbacks of the NSGA is that this algorithm compared

to the MOGA and NPGA is competetationally expensive it has been observed that a computational time that is required for this is order of m n cube if you consider the entire procedures there.

So, other than this inefficiency so for that timing or the time that is the time that is required to compute this algorithm is concerned, there is another is that this algorithm should consider one parameter to be decided by the programmer, it is here in order to niche count. So, these are the 2 serious drawbacks, that is there in this NSGA and another approach is that this is basically non elitism approach, because it gives a favor to some times the inferior solution also, because the non lambed front is basically or the front we have considered they are basically the non elitism approach.

So, these are the 3 I mean critism against the NSGA, and then the same author who proposed NSGA, they have proposed another person of the NSGA, it is called the NSGA 2 algorithm. So, we will discuss this NSGA 2 algorithm in the next lecture.

Thank you.