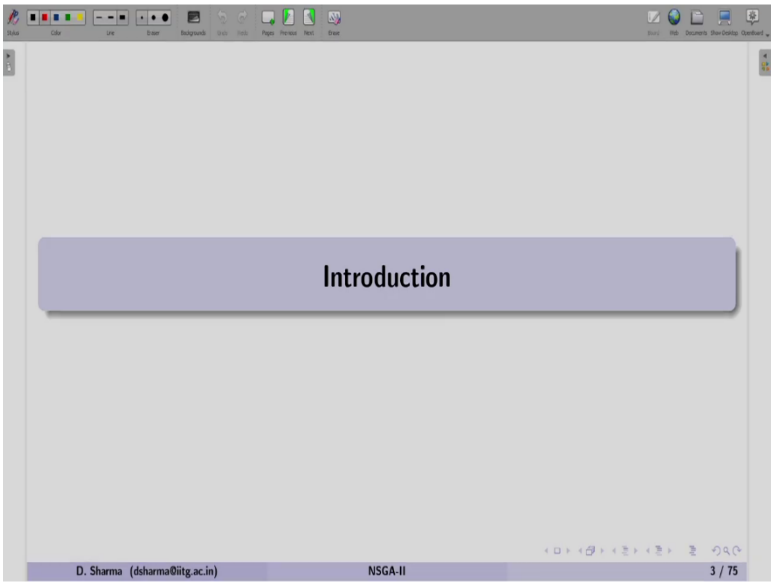
Evolutionary Computation for Single and Multi-Objective Optimization
Dr. Deepak Sharma
Department of Mechanical Engineering
Indian Institute of Technology, Guwahati

Lecture - 21
EC Techniques for Multi-Objective Optimization: NSGA-II

Welcome to the session on EC Techniques for Multi Objective Optimization. Till now, we have gone through the introduction of multi objective optimization followed by classical methods for the same. In this session, our main discussion will be focusing on one of the benchmark algorithm called NSGA II.

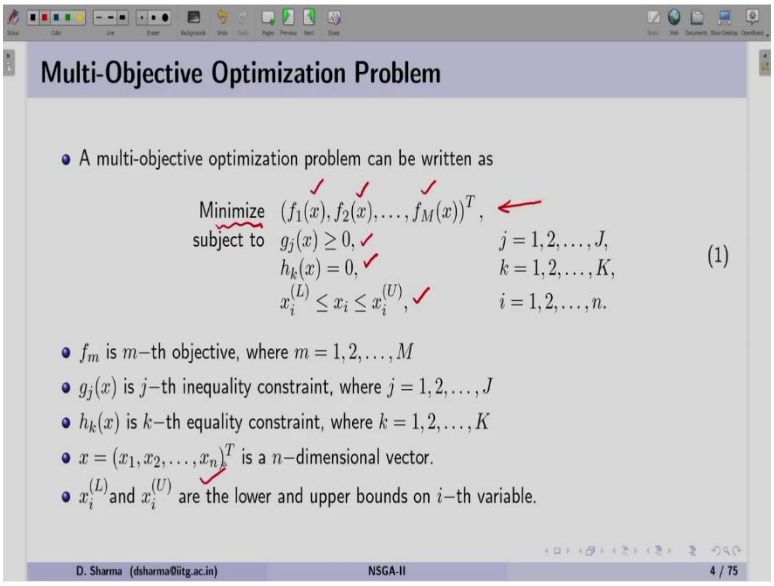
In this session, we will be going through the introduction first thereafter, we will discuss NSGA II. We will take one example and we will understand all the working principles of this algorithm. Thereafter, some simulations on mathematical multi objective optimization problems, we will see the simulation or the results of the same algorithm.

(Refer Slide Time: 01:40)



Let us begin with the introduction now, as we know that a multi objective optimization problem can be written as in equation 1, that we want to minimize the objectives.

(Refer Slide Time: 01:52)



$$Minimize \qquad \left(f_1(x), f_2(x), \ldots, f_M(x)\right)^T,$$

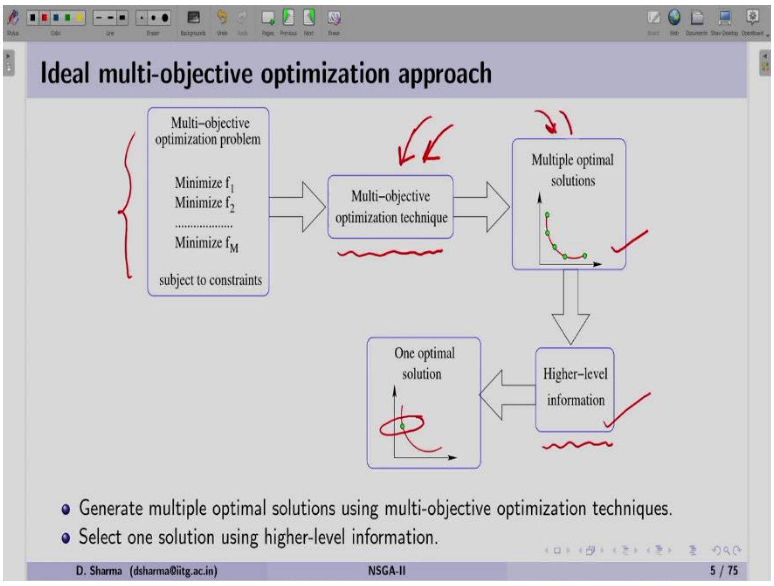
$$subject\ to \qquad g_j(x) \geq 0, \qquad j = 1,2, \ldots, J$$

$$h_k(x) = 0, \qquad k = 1,2, \ldots, K,$$

$$x_i^{(L)} \leq x_i \leq x_i^{(U)}, \qquad i = 1,2, \ldots, n.$$

So, instead of one objective we have multiple objectives. All these objectives are conflicting in nature, and this problem is subjected to various constraints such as inequality constraint, equality constraint and the variable bounds. Here, we have a vector of m conflicting objectives and we also have a vector of the decision variables.

(Refer Slide Time: 02:32)

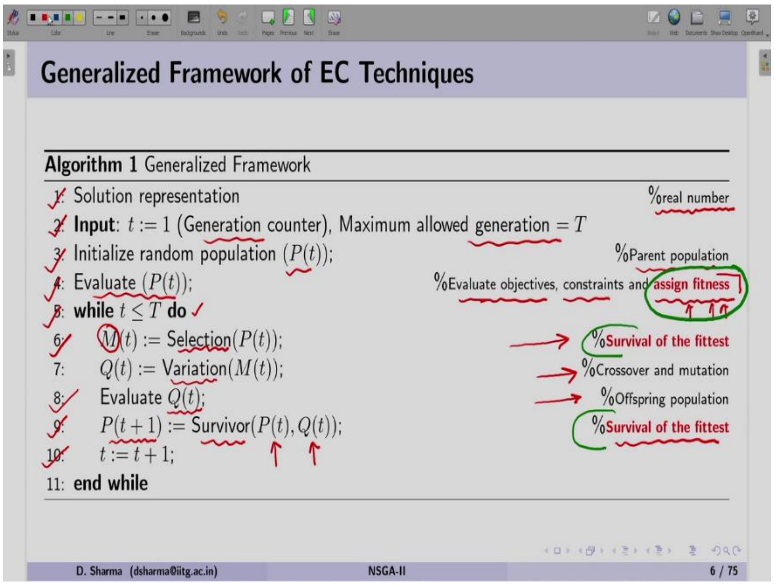


Since, we are familiar with this particular multi objective formulation. So, we know that there are two approaches to solve the given problem. The first approach is the preference based approach, in which the classical multi objective optimization methods solve or we can say convert the multi objective problem into a single objective problem. Those methods use the higher level information to convert this problem into a single objective problem.

Then we can use any single objective optimizer to get a solution. The ideal approach the second approach as we know, that in this particular approach we solve a multi objective problem by the method or a technique that can generate the multiple optimal solutions and those solutions will be lying on the Pareto front.

Once the solutions are generated the higher level information can be used, and that information will help us to find the final optimal solution that we are going to design to for designing and for some decision making. As we know that EC techniques are population based techniques; so, these techniques can very well fit into the ideal multi objective optimization approach that can generate the multiple solutions on the Pareto optimal front.

(Refer Slide Time: 04:28)



So, with this introduction we have till now going through the one algorithm to the another algorithm, using a generalized framework. If we look into the same framework, we can see what are the places we can change, so the same framework can be used for multi objective optimization.

So, if we see the algorithm number 1 in this case at step 1, we perform the solution representation. So, let us assume that we want to target the real number means real parameter multi objective optimization problem. In step 2 there are various input, such as the generation counter, the maximum allowed generation and the other inputs depending on the operators we are going to use it.

3rd is we are going to initialize the random population, and this population we are referring as a parent population. So, meaning that the variable x 1, x 2, x 3 we are generating randomly within the bound. In step 4 we evaluate the population meaning that we are going to evaluate the objective function the constraints.

And, now we have to think about that what should be the fitness assignment so, that we can use the generalized framework for multi objective optimization. So, we remember that here instead of one objective, we have multiple objectives. And therefore, the fitness assignment is going to change for multi objective optimization. Now, in step 5 we are in the standard loop of number of generations now currently t is smaller than T. So, we are going inside.

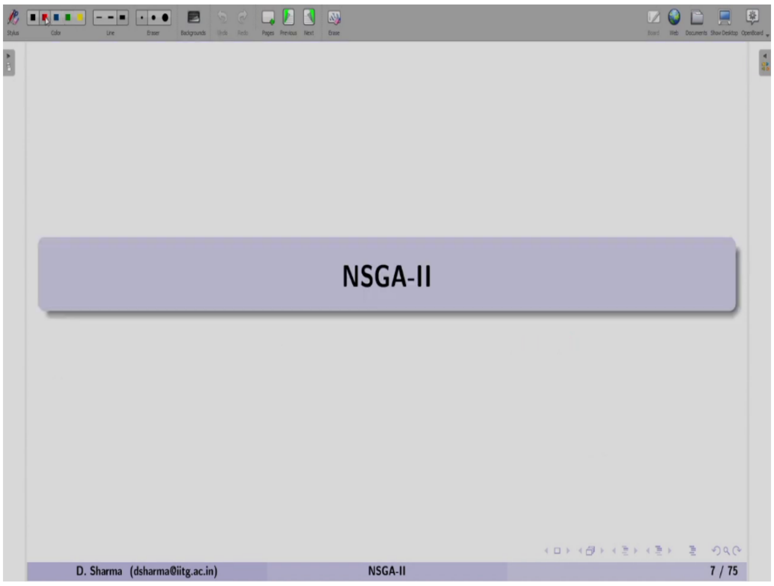
And the step number 6 says that the we have to do first is the selection, to create a mating pool which is represented by M. Now, here although we have assigned a fitness to each member of the population; now, we have to see that since the problem it is in multi objective so how we are going to use the survival of the fittest.

Once it is done then we apply variation, basically operators similar to crossover and mutation to change the members or their decision variable vector; after applying those operator since the population is new or the members in the population are new so, we have to evaluate. Evaluate again is objective function, constraint and assigned fitness. And this population we remember that it is referred as a offspring population.

Now, in step number 9 we have to make the population for a next generation using the survivor. Here we have parent population, we have offspring population. So, this so we are going again to use some kind of a survival of the fittest here to choose it. And then we increase the counter by 1 and then this loop will be followed till the termination condition get satisfy.

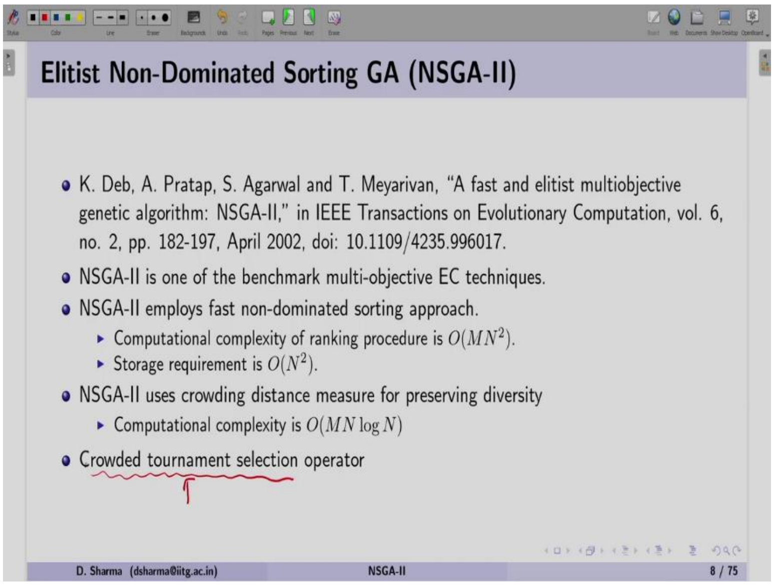
So, from this particular frame work, we can see that if we target our focus on fitness assignment. Similarly the survival of the fittest at two stages, then the same framework can be used for multi objective optimization.

(Refer Slide Time: 08:44)



So, with this introduction let us understand the algorithm which is known as NSGA-II.

(Refer Slide Time: 08:53)



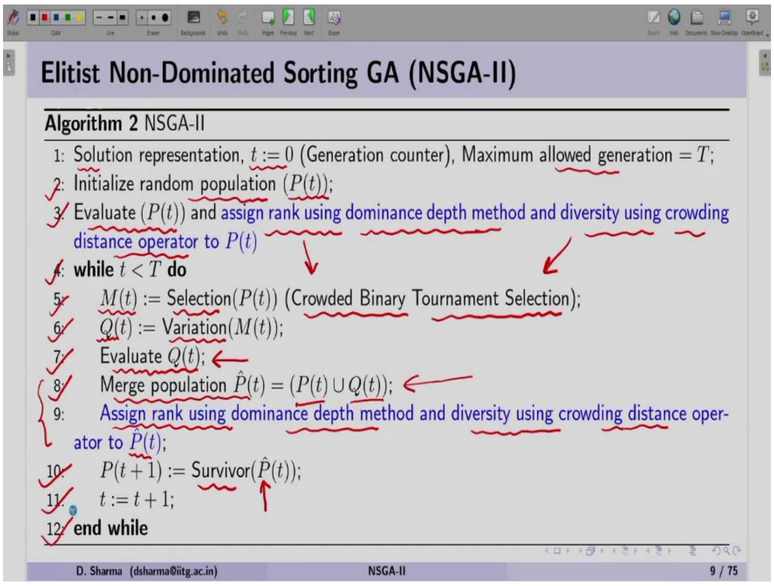
This algorithm is developed by Professor Deb and his co authors, and the algorithm was named as fast and elitist non dominated fast and elitist multi objective genetic algorithm which is NSGA-II. So, here II stands for the part II. So, NSGA stands for Non dominated Sorting Genetic Algorithm.

This algorithm is one of the benchmark multi objective EC technique. There are certain features because of that this algorithm is popular; one of the features is that NSGA-II employees, fast non dominated sorting approach. In this particular approach the computational complexity of ranking procedure is order of M N square and the storage requirement is of order N square.

Now, as we remember this M capital M stands for the number of objectives in our problem, and capital N stands for the population size. So, this non dominated sorting will be useful for the convergence. So, NSGA-II uses crowding distance measure for preserving the diversity. So, this operator has the computational complexity of order of M N log N.

Here again M is the number of objectives, N is the population size. The another good feature of this algorithm is the crowded tournament selection operator. So, the tournament selection operator we know. So, this particular operator was little modified. So, that it will work for multi objective optimization.

(Refer Slide Time: 11:16)



Now, let us see the algorithmic representation of NSGA-II, we start with the solution representation. And, now we have clubbed the input as t the generation counter and the maximum allowed generation. In step 2, we initialize the random population, which is we are referring as a parent population.

Step 3 is the important step, because we evaluate the population meaning that we calculate all objectives and constraints. And, thereafter we have to assign a rank using dominance depth method and the diversity using crowding distance operator. So, this dominance depth method, we discussed in the previous sessions that this particular method makes different fronts such as front 1, front 2, front 3 etcetera and it is sorting the solutions in different fronts.

We will again go through this dominance depth here. And, we also we will also discuss the crowding distance. Thereafter, in step number 4 we are in the standard loop of the number of generation or the generation counter.

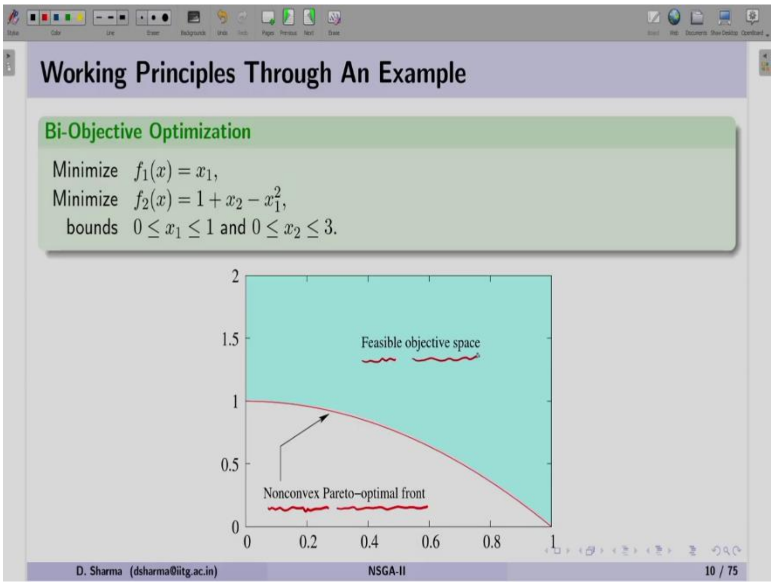
Step 5 we remember that it is the selection. So, in NSGA-II the crowded binary tournament selection is used. So, it is a modified version of binary tournament selection operator so, that we can incorporate both the rank as well as the diversity into the selection.

Once the selection is done and mating pool of M t is prepared, we will be performing the crossover and mutation and this crossover and mutation will be changing the members and generate the offspring population called Q t. Thereafter, in step 7 we evaluate the offspring population. Again evaluate the offspring population means, we will be calculating all the objectives and the constraints.

Thereafter, in step 8 we merge so, here we can see we are merging the population. So, the parent population and the offspring population both the populations are merged. And, thereafter the fitness the rank is assigned using the dominance depth method, and diversity is assigned using the crowding distance operator on this merged population.

So, here that is a small change you can find that instead of just ranking the new population Q t, here both the populations are combined in step number 8. And, then rank and the diversity both are calculated for the combined population. Thereafter, in step 10 we have the survival that will be applied on the combined population of parent and offspring. In step 11, the counter is increased by 1, and in and this particular while loop we follow till the termination condition gets satisfied.

(Refer Slide Time: 15:21)



$$Minimize \quad f_1(x) = x_1,$$

$$Minimize \quad f_2(x) = 1 + x_2 - x_1^2,$$

$$bounds \quad 0 \le x_1 \le 1 \quad and \quad 0 \le x_2 \le 3.$$

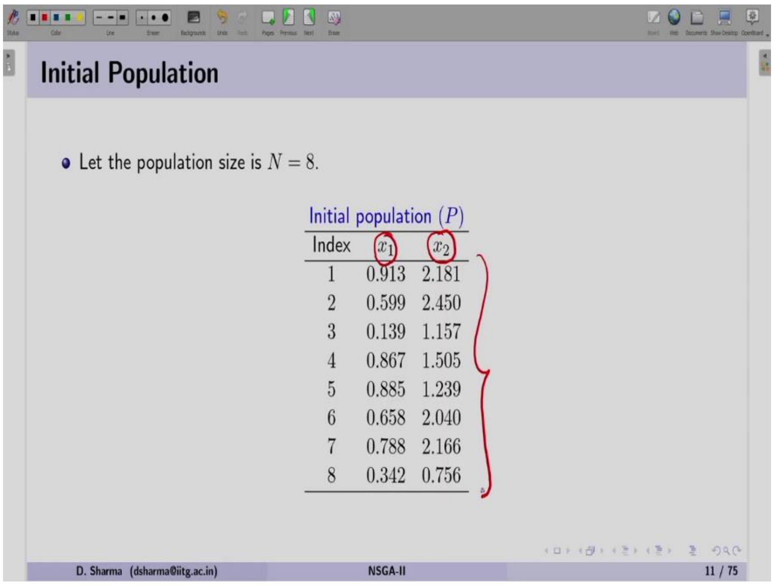
So, now this is the algorithmic representation of NSGA-II, this is the main algorithm. We are going to understand this particular algorithm with an example. So, the working principle in the terms of steps we have understood this we will be applying on a small population for the given example.

So, the example that we have chosen is a bi-objective optimization problem bi means we have two objective problem, we want to minimize the first objective f 1 x is equals to x 1. And minimize f 2 x which is 1 plus x 2 minus x 1 square. The range of x 1 is 0 to 1 and the range of x 2 is 0 to 3.

If we look its Pareto optimal front, so we can see that we are going to solve a problem which is having a non convex Pareto optimal front. And above this region as we have mentioned in the figure the space is the feasible objective space. Now, we remember that for the classical multi objective optimization methods, many of the methods cannot be used for the problems which have the non convex Pareto optimal front.

But, this such kind of a limitation does not exist with the EC techniques, and they can find the Pareto optimal solution for any kind of a problem. Therefore, we chose these two objective non two objective problem which has a non convex Pareto optimal front.

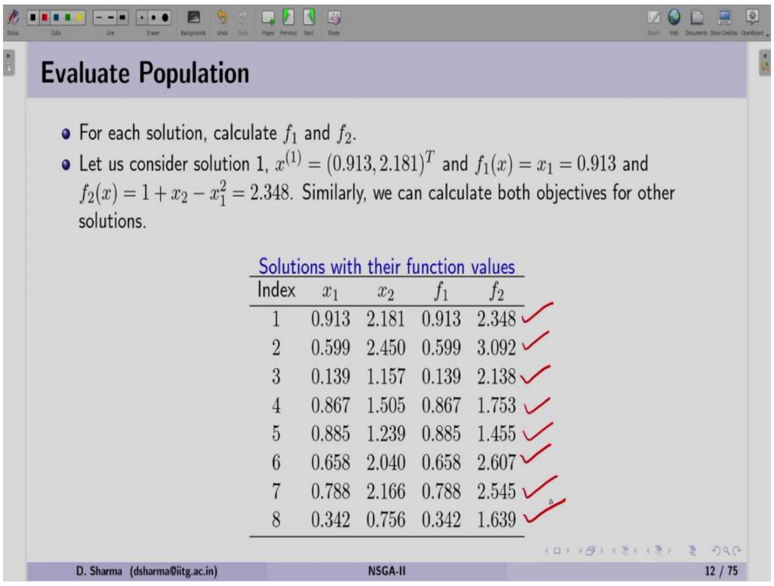
(Refer Slide Time: 17:20)



Index	x_1	x_2
1	0.913	2.181
2	0.599	2.450
3	0.139	1.157
4	0.867	1.505
5	0.885	1.239
6	0.658	2.040
7	0.788	2.166
8	0.342	0.756

So, as per our algorithmic representation so, the first step is we have to initialize the population. Let us assume that the size of the population is 8. Now, here when we are generating the initial population meaning that we are generating x 1 and x 2 randomly so, within the range of x 1 and x 2 these solution; so, 8 solutions are generated randomly.

(Refer Slide Time: 17:51)



Index	x_1	x_2	f_1	f_2
1	0.913	2.181	0.913	2.348
2	0.599	2.450	0.599	3.092
3	0.139	1.157	0.139	2.138
4	0.867	1.505	0.867	1.753
5	0.885	1.239	0.885	1.455
6	0.658	2.040	0.658	2.607
7	0.788	2.166	0.788	2.545
8	0.342	0.756	0.342	1.639

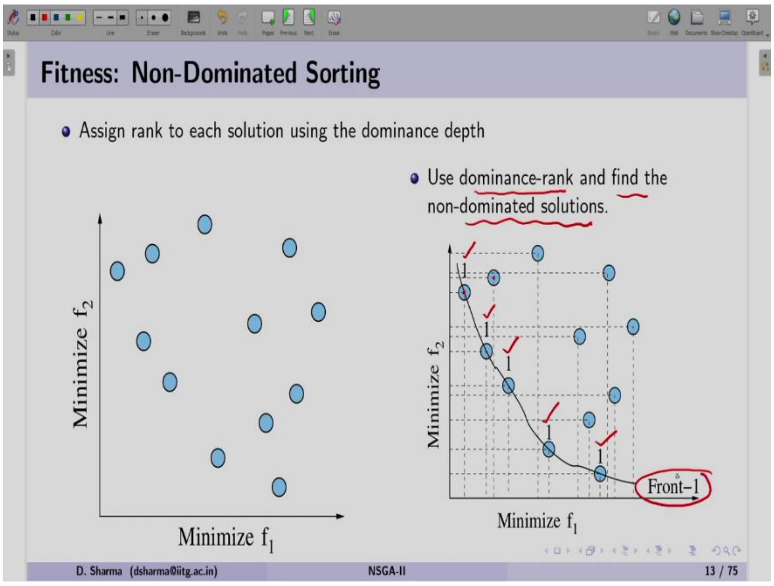
The next step is the population we have to evaluate the population. So, for each solution we will be calculating the f 1 and f 2, in the given problem there is no constraint. So, we are just focusing on f 1 and f 2. Let us consider solution number 1, in this solution as in the previous slide we understood that the first solution having two component as x 1 and x 2 we can see here.

In order to find the value of f 1 we will be including their values. So, that I can get f 1 similarly f 2 including the value of this x 1, x 2 component of solution 1 we get 2.348. Now, similarly we can calculate both the objectives for other solution, because it is very straight forward.

$$x^{(1)} = (0.913, 2.181)^T \; and \; f_1(x) = x_1 = 0.913 \; and \; f_2(x) = 1 + x_2 - x_1^2 = 2.348$$

So, once we are including the value of x 1 and x 2 in f 1 and f 2, we can see that for all the solutions we have their f 1 and f 2 values. So, these f 1 and f 2 values for every solution will help us to find out their rank and diversity.

(Refer Slide Time: 19:16)

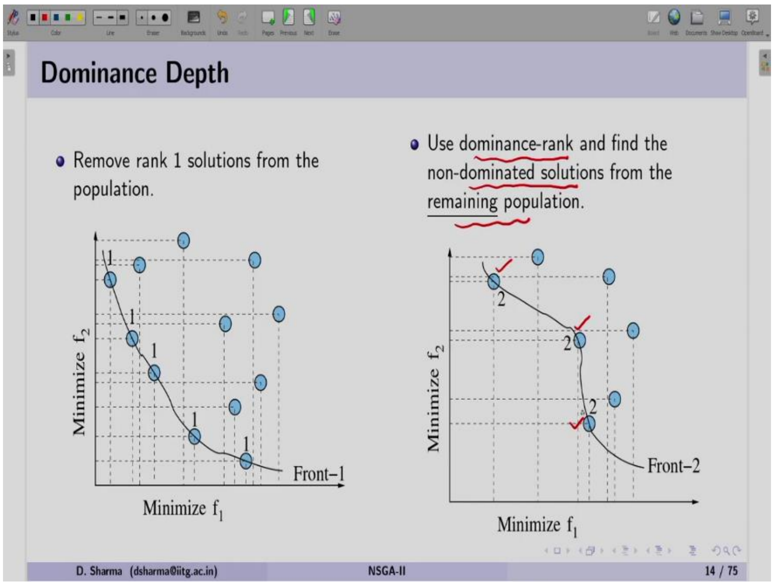


Now, as of now we have calculated the objective function. Now, we have to assign the fitness to each solution. This particular fitness will be assigned using Non Dominated Sorting method and that is why it is called NDS. Here we are we will assign rank to each solution using the dominance depth.

So, this dominance depth method we already discussed. So, as a recap let us understand how this dominance depth method works. So, let us see the case when we want to minimize f 1, we minimize f 2 and there are certain solution in this objective space. Then, we use dominance rank and find the non dominated solution.

Meaning that we will be using the concept of dominance and we will be compare each and every solution shown in the figure, while applying the dominance rank we get to know that the solutions where the 1 is written all of them are rank 1.

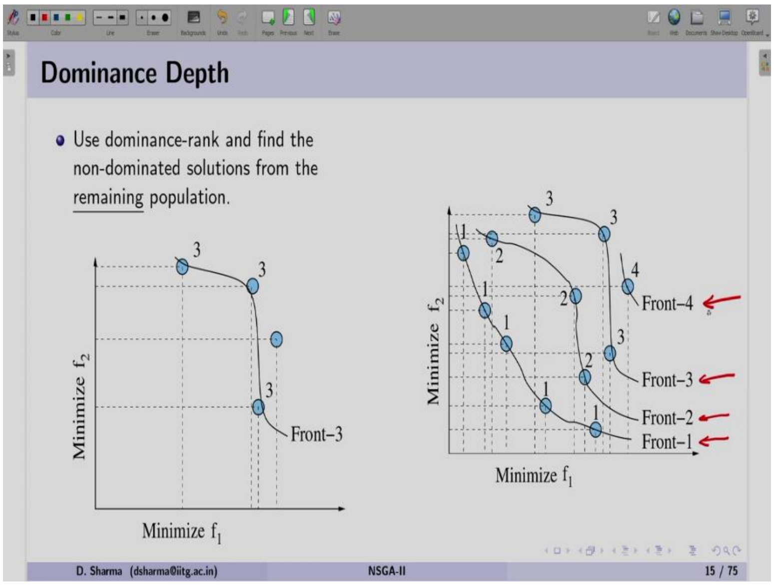
(Refer Slide Time: 20:41)



So, these solutions are copied into front 1, thereafter once it is done then we remove the these solutions. So, the front 1 solution we are going to remove it, when we remove these are the remaining population with us.

Then, again we are going to apply the dominance rank to find out the non dominated solution in the remaining population. So, there we will find the solution where 2 is written. So, these solutions are non dominated, the now their rank is 2 and they are copied into the front 2.

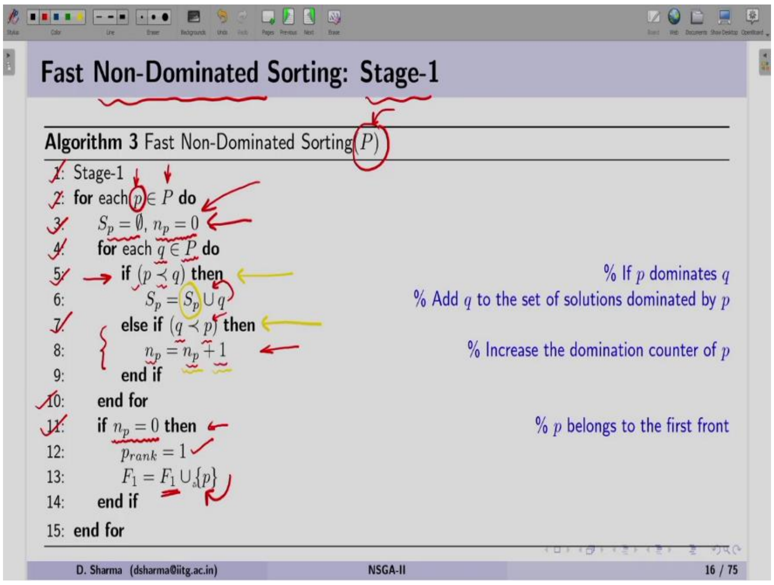
(Refer Slide Time: 21:18)



Similarly, we will remove the solution of rank 2 as well and we are left with these solutions. Again if we find the non dominated solutions. So, they will become the solutions where the 3 is written. So, the solutions having a rank 3 and they are copied into front 3. And finally, all the solutions the dominance depth method we will sort in the front 1, front 2, front 3 and front 4 for the given example.

Now, while following the dominance step method, we can see that if we are going to find the worst complexity of this method, this is going to be order of M, N, Q. So, M is the number of objective, N is the number of population number of members in the population. So, since it is computationally expensive. So, NSGA-II came up with the idea of fastest non dominated sorting that we are going to discuss now.

(Refer Slide Time: 22:32)



This fast non dominated sorting, we will be presenting in 2 stages. So, the first stage is as we can understand from the algorithmic representation, this is stage 1; in step 2 we have say member p which belongs to our population P. So, the input to this non dominated sorting is the P as we can see on the top.

So, every member in the population we will find a set called S p. So, this S p is currently empty, in step 3 and there is a number n p which is also currently 0. So, we will see the significance of these two parameters that we are going to use here. Now, we will take the another solution. So, if we look at step number 4, we are going to take another solution q which is also a part of the population same population.

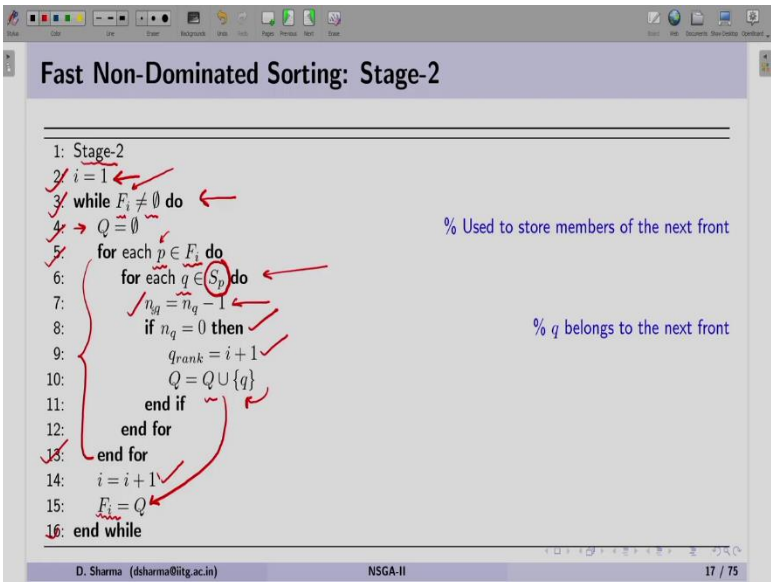
Now, let us compare these two solution. So, in step number 5 if p is dominating q, then we will be adding q into S p meaning that S p set consist those solutions, which are dominated by p. In step 7, if suppose solution q is dominating p, in this case we are adding n p is equals to n p plus 1, meaning that this counter will increase by 1 if any solution in the population is dominating p.

So, if we see again the difference between the step number 5 and step number 7, step number 5 they says that p is dominating q. And if it is happening the solution is saved in S p, in step 7 the solution q is dominating p in this case the n p which is corresponding to the member p that is increased by 1.

Now, up to step 10 we have compared all the solution in the population P with the small p which is given in step number 2. Here you can see that we are not comparing p versus p, thereafter we will look is there any solution which has n p equals to 0. So, what is the significance of n p equals to 0?

So, if we look back the step number 7 here, in this step it says that if any solution q is dominating p, then only we will increase the counter of n p. If it is not then n p which we have initialized as 0 initially that will remain the same. So, at step 11 if we are looking for any member n p which is equals to 0 meaning that we are looking for the non dominated solution. Once we identify we will give the rank 1 to those solutions. And we will copy those solution into F 1. So, F 1 will become the front 1.

(Refer Slide Time: 26:29)



So, this is stage 1; now in stage 2 we are not going to perform the non dominated sorting or any kind of a ranking again. So, let us see how it is done. In step 1, we have a stage 2 and at step 2 we are assigning i equals to 1. Step 3 this is a while loop and the condition is F i is not a empty set. So, currently i equals to 1 this means we are talking about F 1. Now, since it is not empty from our previous stage, stage 1 so, it will go inside the while loop. In step number 4, there is a set called Q which is currently empty.

Now, in step number 5, we will take all the solutions that belongs to F i. So, currently it is front 1. Now, for all those solutions p which belongs to F i, we will be finding what is their S p. So, meaning the solutions that are dominated by p. So, we are making q belongs to the set of S p and for the solutions which are lying in S p we are subtracting this num n currently it is n q is equals to n q minus 1.Aat this particular stage, if any of the solution having n q equals to 0, we are going to assign a rank i plus 1.

So, currently i equals to 1 so, the rank will become 2 and the solution will be copied into Q. This process will be repeated till step number 13. And thereafter what we have done we have taken all the q belongs to the S p in step number 6 and then after finishing this we are increasing the i equal to i plus 1 meaning that we will be looking for the front 2 now. So, i was 1 earlier now i will become 2.

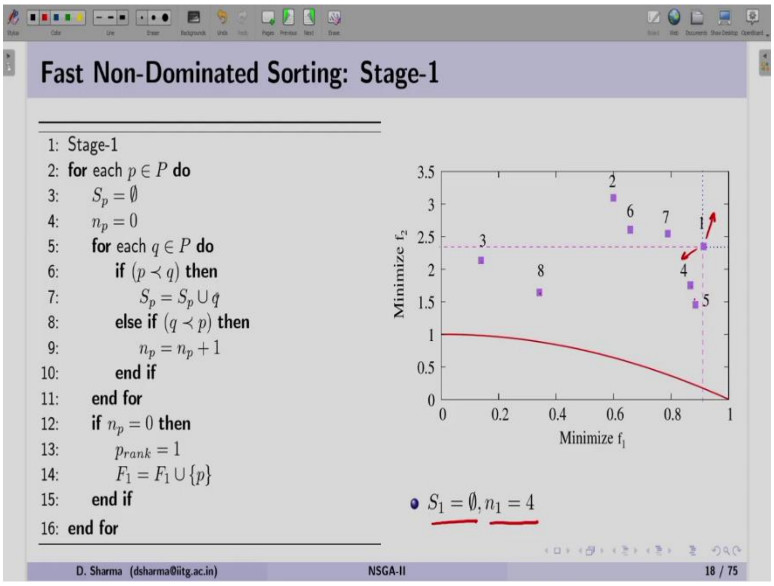
Now, since i equals to 2 in front 2 we are going to save all the component of a Q into F i. Now, again in step number 16, we will go back to the step number 3. If there is any member in front 2, we will be following the same steps. So, from this particular discussion, what we understood is that in the stage 1, we performed the dominance of a concept.

And using that concept we try to find out what are the solutions that are non dominating. Along with that we have used two parameter such as S q or S p and n p these two parameters take cares of when the solution p is dominating q. So, q will be added into S p and if p is dominated by q, then the counter of n p is increased by 1.

So, that way we have used the concept of dominance for the first time. In the stage 2, we are using the sets such as S p and n p to find out what will be the second front solutions in the second front and similarly third front. So, we have eliminated again using the concept of dominance for the rest of the members so that we can find the solution in front 2, front 3 and so on as we have discussed in the dominance depth method.

Now, this particular these two stages we will be understanding with the given example, where we generated the 8 solutions randomly and we have already calculated their F 1 and F 2 values.

(Refer Slide Time: 30:46)



So, let us take the same let us take the solutions. So, in the right hand side all the major steps of stage 1 are written here. Let us look at the solution on the right hand side. So, the solutions and the marking such as 1, 2, 3 etcetera that are being used so, that we can see where is the solution 1 in the objective space, similarly solution 2 in the objective space.

Now, in our previous discussion when we use say concept of a dominance, use for solution 1 meaning solution 1 will be compared with the rest of the members in the population and we will find out the solution 1 is dominating which solution and solution 1 is dominated by what are those solutions.

So, since it is a two objective problem and both of them are minimization. So, we remember that if we take solution 1 as our reference. And if we look into the first quadrant so, that is the right hand side top corner. So, if there is any solution which is lying in this particular rectangle. So, then these solutions are dominated by 1.

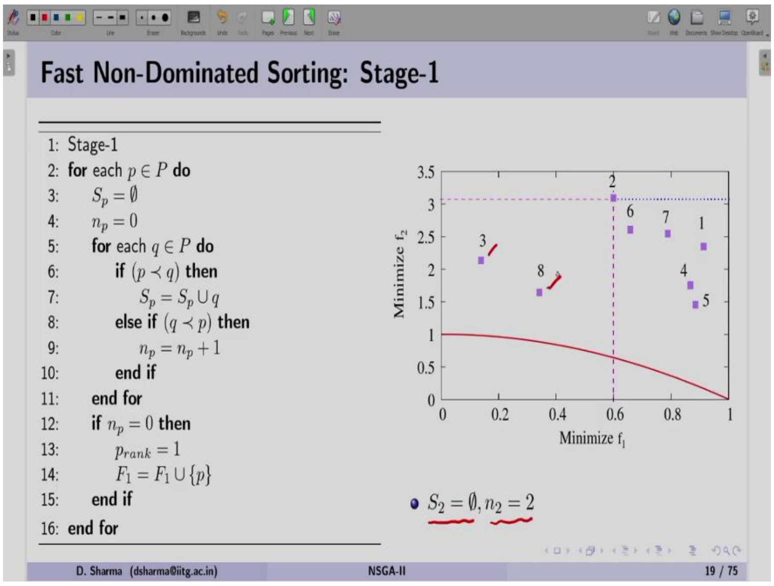
So, current status says that there is no solution which is dominated by 1. Similarly, we have we also made our understanding for two objective problem that if we take a solution 1 as our reference and, if we look into the third quadrant. So, the solutions which will be lying in the third quadrant are those solutions that are dominating solution 1. So, looking at the figure we can see that solution 3, solution 4, solution 5 and 8 all are dominating 1.

So, in this case we have this step number 3 where we are assuming S 1 and n 1 both are 0 S 1 is empty set. Now, 1 will be compared with rest of the members. So, in this case when we are applying as per our discussion S 1 is empty set, because there is no solution that is lying in the first quadrant with respect to solution 1.

However, if we look at the third quadrant there are 4 solution 3, 8, 4 and 5. And therefore, n 1 is 4. So, we are calculating the step number 7 and 9 here. Let us take another solution now that is solution number 2. Again using the similar approach, we will look into the first quadrant.

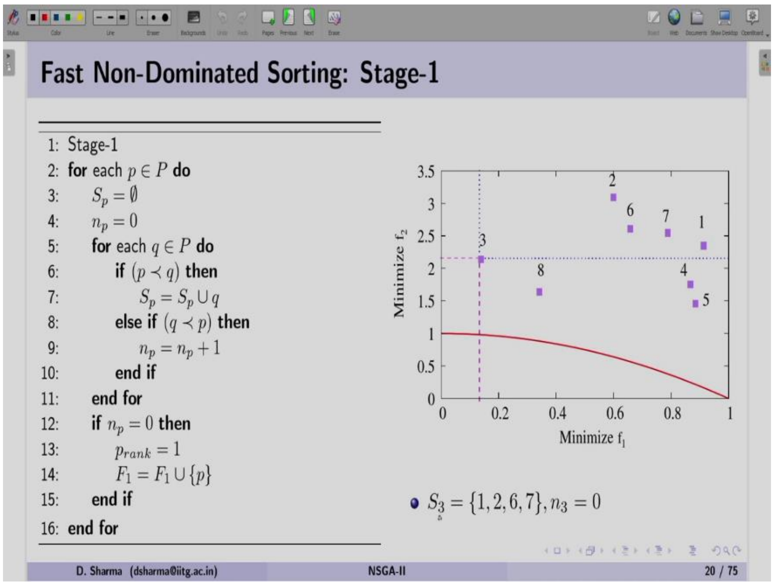
In this first quadrant with respect to solution 2 there is no solution meaning that solution 2 is not dominating anyone. If we look the third quadrant with respect to solution number 2, we have two solution 3 and 8.

(Refer Slide Time: 34:11)



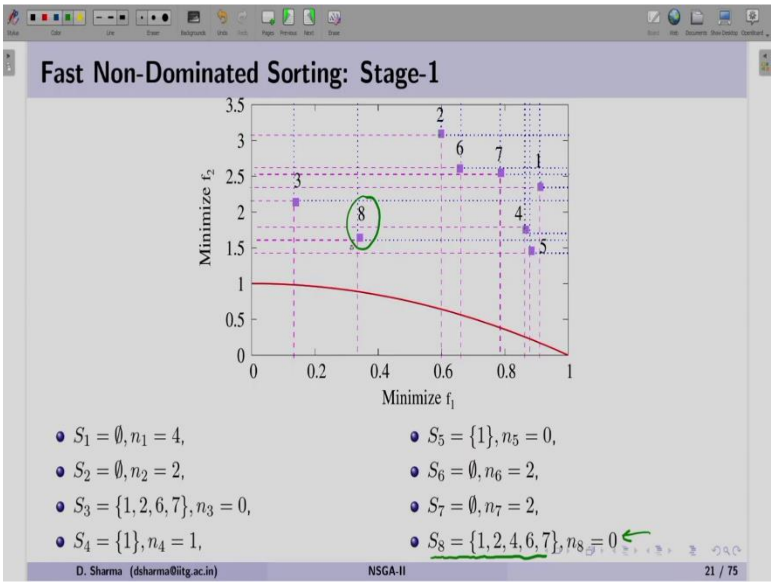
Therefore, if we look S 2 this is empty for solution 2 and n 2 is 2, because it is dominated by the two solutions. We will take another case where we have chosen solution number 3. Now, if we take solution 3 as our reference looking at the first quadrant. Now, we can see solution one is dominating many solution, such as 1, 2, 6 and 7 also. If we look at the third quadrant of the solution 3 there is no solution.

(Refer Slide Time: 34:59)



So, by following the step number 7 and 9 here in the stage 1 we will get to know that S 3 is now made of 1, 2, 6, 7 why because these are the solutions dominated by solution 3; n 3 is equals to 0 meaning that the solution 3 is not dominated by any of the solution in the current population.

(Refer Slide Time: 35:24)

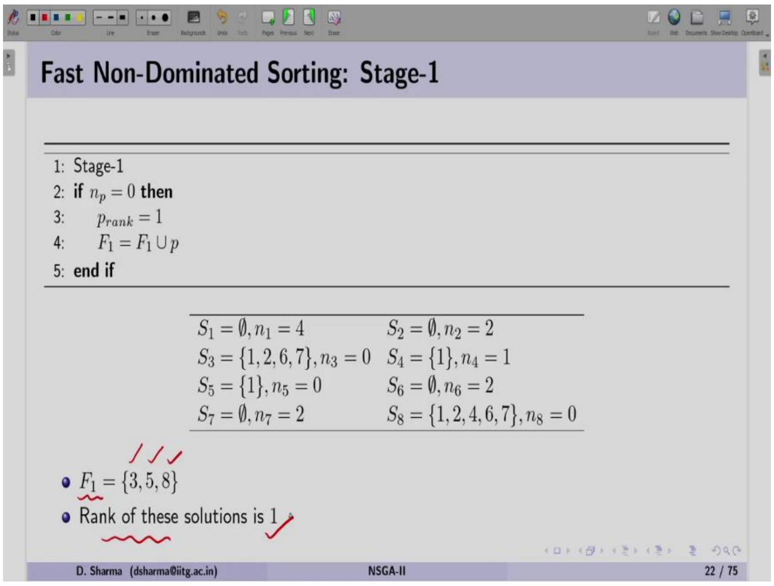


So, therefore, if we follow the same procedure for all the solution so, we can see that solution 4 is here. And looking at the first and the third quadrant, we can see S 4 is 1 and n 4 is 1 because 4 is dominated by 1. Similarly, we can look for solution 5. So, the solution

1 is dominated by 5 and n 5 is 1 means it is a non dominated solution. Similarly, if we look at solution number 6, so the S 6 is a empty set meaning it is not dominating any solution and n 6 is 2 meaning it is dominated by solution 3 and 8.

Let us take in the case of solution 7 here S 7 this should be S 7 it is a empty set, because it is not dominating any solution and n 7 is 2, because it is dominated by solutions 3 and 8. Finally, let us take solution number 8. Now, looking at 8 we can see that it is dominating 1, 4, 6, 7 and 2. So, therefore, S 8 is comprised of 1, 2, 4, 6 and 7 and n 8 is 0, because there is no solution which is dominating solution number 8.

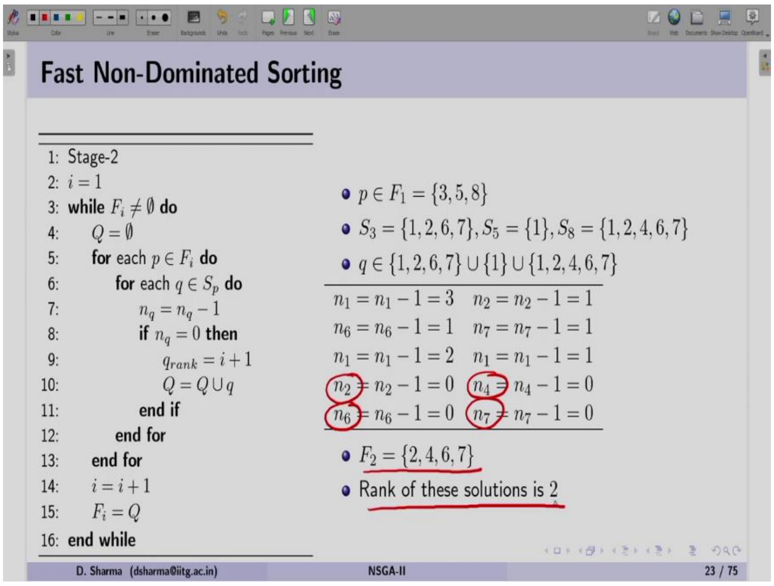
(Refer Slide Time: 37:02)



So, if we find, which are the non dominated solution? So, as per our stage 1 here, we will be looking for those solution which has n p equals to 0, we will be assigning the rank 1 and adding those solution in front 1. In this particular table, we can see that we have all 8 solutions there S p and n p are written. Looking at that value, we can see the solution number 3, solution number 5 and solution number 8, they have n p equals to 0 meaning that these solutions will be added into front 1.

So, as we can see they are assigned in F 1 and rank of the solution will become 1. So, this is the hand calculation for stage 1. Now, we will see how we can use the information that are saved in S p in terms of S p and n p for stage 2.

(Refer Slide Time: 38:19)



So, let us look here now in this stage 2 as we remember in step 2 i equals to 1. So, basically we are looking for front 1. Now, the front 1 so, the in step number 5 we see that the p belongs to F 1. So, here on the right hand side p belongs to F 1 are 3, 5, 8. Now, we have to find out what are their S p. So, here S 3 is consist of 1, 2, 6, 7; S 5 consist of 1 and S 8 consist of 1, 2, 4, 6, 7.

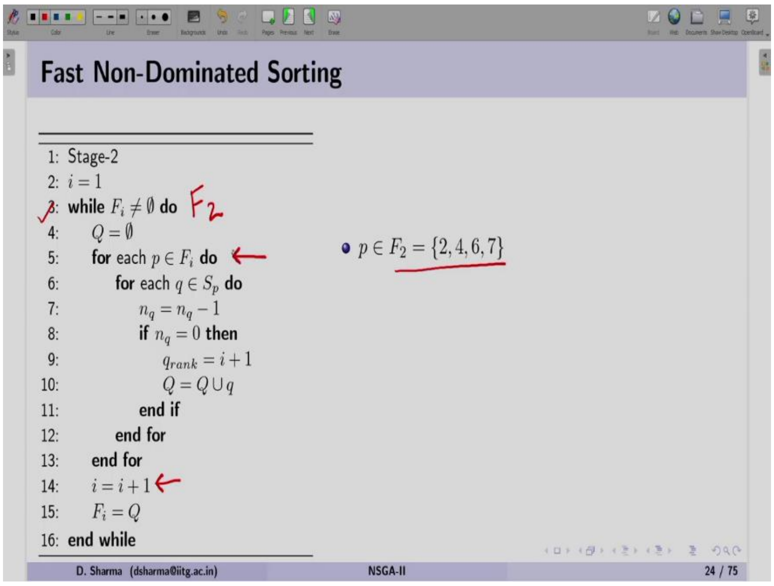
Now, if we are going to find the q which is as per our step number 6. So, q will belongs to 1, 2, 6, 7 union 1 and then union 1, 2, 4, 6, 7. So, these members we are going to take one by one. So, as suggested in step number 7 we will be subtracting n q equal to n q minus 1. So, let us take the first in the set which is solution number 1.

So, we are subtracting n 1 equal to n 1 minus 1. So, here the earlier value was 4. So, n 1 minus 1 will become 3. The next solution here is solution number 2 and in this case the n 2 is equals to n 2 minus 1 is equals to 1 now. The third solution here is 6. So, we will take n 6 equal to n 6 minus 1 we get 1 and then we have 7.

So, n 7 equals to n 7 minus 1 which is now 1. Thereafter we will take 1 so, n 1 is now 2 then we will take again 1. So, the n 1 value is updated to 1 now. Now, we will take solution number 2 if we take 2. So, we get n 2 equals to 0, then we will take 4, n 4 is equals to 0. Thereafter we will take solution 6, n 6 is equals to 0. And finally, solution number 7 so, that is also 0.

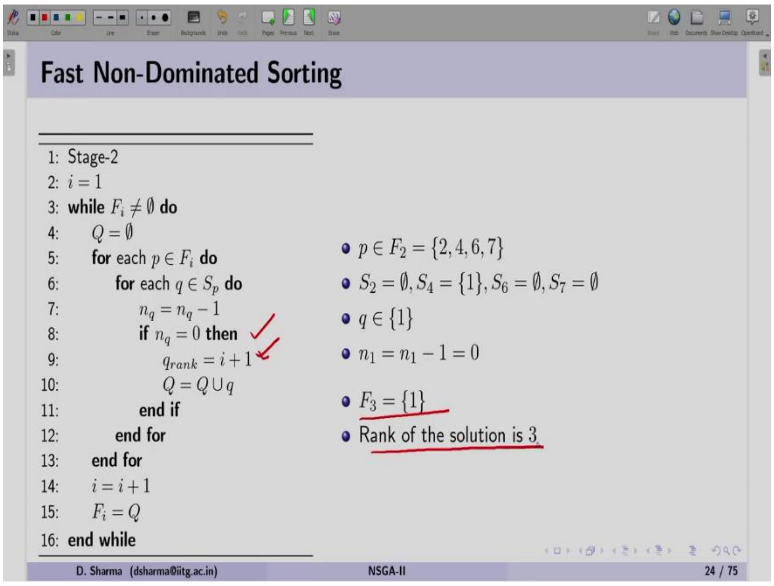
Now, if we look at the step number 8 on the left hand side, it says that if there is any solution which has n q equals to 0, then we are going to assign a rank i plus 1. So, since i is equals to 1 so, this will become rank 2. We will be adding this q into the capital Q so, if we look on the right hand side table, we can find n 2, n 6, n 4 and n 7 all of them are 0 meaning that they are going to be added into the front 2 and the rank of the solution will become 2.

(Refer Slide Time: 41:53)



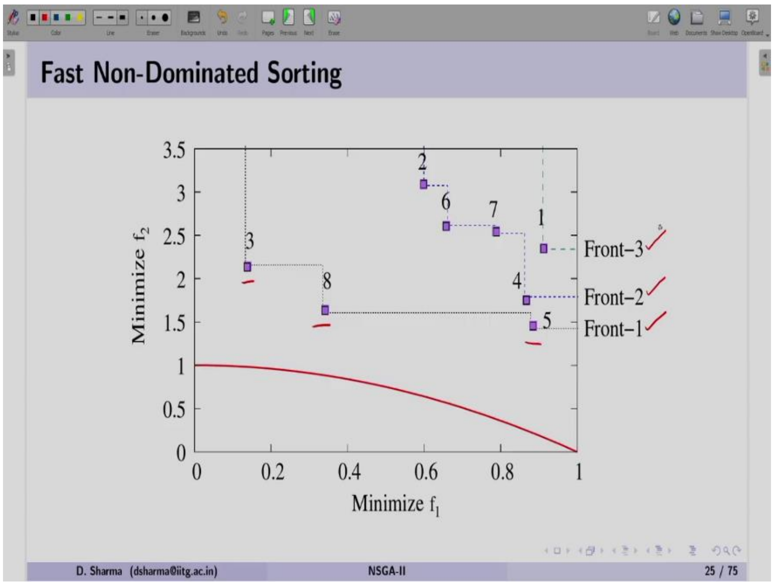
Now, as of now we have identify front 2, now we are again going to step 3. So, as per our previous slide we are going to increment i plus 1 so, this i will become 2. So, in this case we are dealing with front 2. So, just now we calculated front 2 and the front 2 members are 2, 4, 6 and 7. So, that belongs to the step number 5.

(Refer Slide Time: 42:25)



Thereafter, we will find what is S 2 then S 4, S 6 and S 7; so, since this is empty set 6, 7, 2, 6 and 7. So, q will belong to just one solution. So, there we will be using n 1 equal to n 1 minus 1 which is 0 and as per the; as per the step number 8 and 9. This particular solution will be copied into front 3 and the rank of the solution is 3 now.

(Refer Slide Time: 43:08)

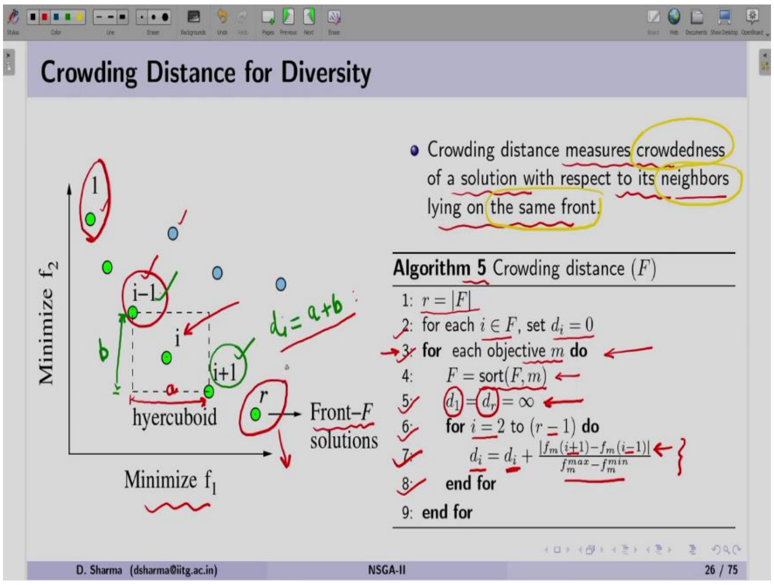


So, as of now we have sorted 3 solution in different front such as front 1, 2 and a 3. So, let us see how these solutions are arranged in the objective space. So, the here figure shows all these fronts, now in front 1, 3, 8 and 5 are the member of front 1. Similarly for front 2, 2, 6, 7 and 4 are the member and solution 1 belongs to the front 3.

We remember that for solving any multi objective optimization problem, there are two goals first goal is the convergence, another goal is the diversity. Non dominated sorting will help us to identify the good solution average solution or bad solution, those solutions are sorted into the different fronts.

So, this non dominated sorting will help us to in the convergence or improving the convergence of the algorithm. So, the rank assigning the rank, associating the solution with the fronts will help us in the convergence. Now, the question comes how we can preserve the diversity among the solution? So, NSGA-II brings the idea of crowding distance.

(Refer Slide Time: 44:48)



So, what is this crowding distance? So, this the crowding distance we are going to use for the diversity. Let us look at the solutions, now purposely we are using different color codings such as green and the blue color. As of now we are considering the solutions all the solutions which are lying in the front F. So, in this solution all the green color dots are belong to the front F. So, their rank is also F.

Now, the purpose of this crowding distance is this is going to measure the crowdedness of a solution, with respect to its neighbors lying on the same front. So, in this particular statement there are two important points, first is we will be looking for the neighbors. And there those neighbor should lie on the same front. And, once we identify then only we can measure the crowdedness of the each solution lying in the same front.

So, the algorithm 5 is given for the crowding distance. So, let us see how we calculate. In step 1 we find out how many solutions belongs to F. So, this is called the cardinality of F so, small r represents the number of solution. In step 2, we will be finding a solution which is belong to the F and for those solution say distance d i we are making is equals to 0.

So, all the green dots which we can see on the left hand side their first distance is set to 0. Now, in step 3 we will take each objective so, one objective at a time, we will sort the solution based on the objective. It is only because we are looking for the neighbors. In step 5, we will be assigning a very large value theoretically an infinite value to the solution which as 1 and the last.

What is the significance? These solutions are the extreme solution of the front F. So, if we look the figure on the left hand side. So, we can see the solution 1 is extreme and solution r is also extreme, we want to preserve these solutions. So, therefore, in step 5 we are assigning a very large value.

So, that is the; that is the crowding distance equal to infinity. For rest of the member in step number 6 starting from say solution 2 to r minus 1, we will be finding the value of a d i is equals to d i why? Because we are adding we are in step number 3, we are performing for each objective. So, let us take this is objective 1. So, d i equals to d i so this is the objective function of i plus 1 solution and i minus 1 solution.

So, if we see the figure on the left hand side, we are calculating the crowding distance of solution i. Since we have sorted the members based on say f 1 objective. So, we know that solution i minus 1 and solution i plus 1, both these solutions are neighbor to it. Now, here in this case when we are finding the distance means, we are finding a difference between the objective function value of i plus 1 and i minus 1.

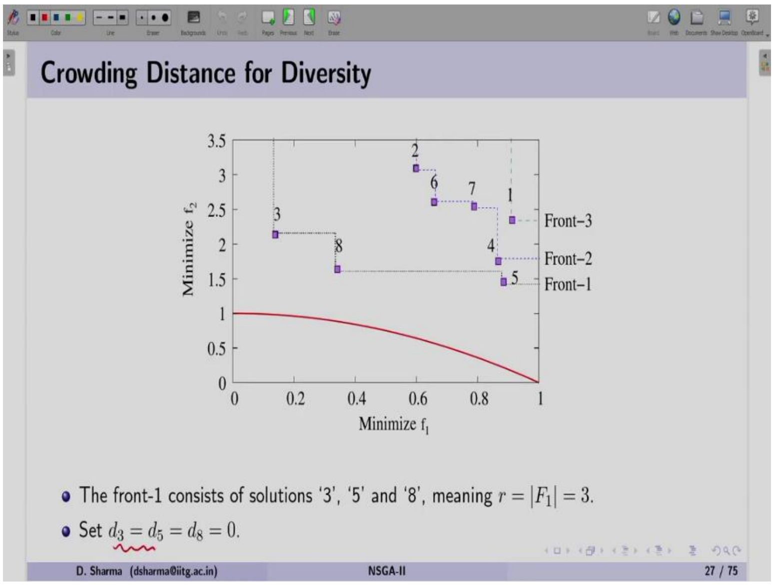
So, basically we are measuring this distance this is along f 1. Suppose the distance this is called a. And here when we find this distance now in step number 7, we are dividing this by f max minus f min of the same objective. It is only because that we can have objectives which may be ranging variedly, meaning that the one objective can be varying in the range of say 10 to the power 6, but the another objective may be varying in 0.1, 0.2, 0.3 range.

In order to unbias our search we do this normalization. So, in this case the difference distance a will be divided by the f max minus f min value. So, in the step number 9 we find the normalized value we added here and then we finish. Thereafter, again we will go to step number 3 for the second objective. Now, when we are going to sort since it is two objective problem i minus 1 and i plus 1 will become the neighbor of the same solution.

In this case when we find the crowding distance for the solution i, let us assume the difference between these two is b. So, the crowding distance of a solution i. So, if we write as d i will become a plus b. Now, in this so similar to this we will be finding the crowding distance of all the solution.

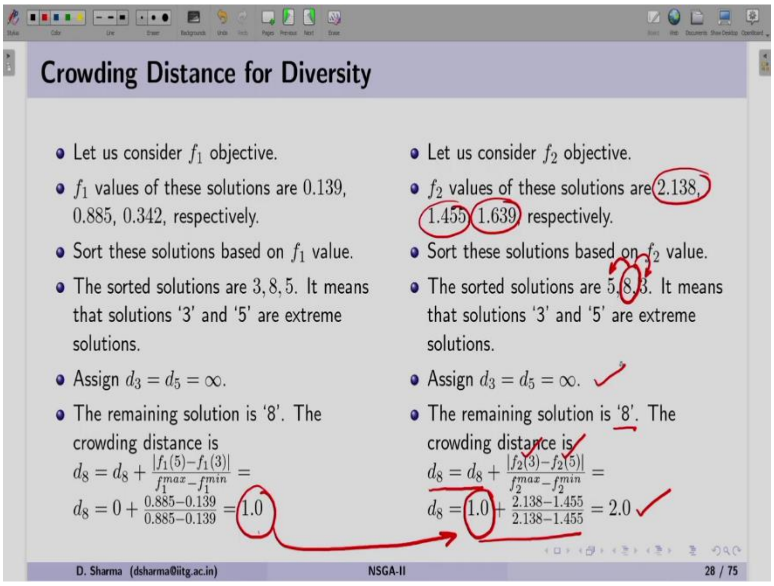
Now, the important point here what we can understand is, this crowding distance which is written as d i equals to a plus b it is nothing but the perimeter. So, this signifies the perimeter of hyper cuboid. So, algorithm 5 represents the how we can calculate the crowding distance for each solution. So, let us apply this crowding distance for the given example.

(Refer Slide Time: 51:32)



Here as we can see in the figure we have front 1, front 2, and front 3 we remember that we apply crowding distance to the same front solution. So, let us take front 1. So, if we take front 1 it consist of solution 3, 5 and a 8 as we can see from the figure. So, the first step is how many solution. So, r is equals to so, the cardinality of F 1 so, we know there are 3 solutions. Initially we set the value of these 3 solution is equals to 0.

(Refer Slide Time: 52:13)



Thereafter, we are going we are going to calculate the crowding distance objective wise. So, let us consider the first objective f 1. For f 1 these 3 solutions have values as we can see 0.139, 0.885 and 0.342. So, the next step says that we have to sort these solution based on their f 1 values.

So, as we know that the minimum value is 0.139 followed by 0.342 and the maximum value is 0.885. So, the solutions are sorted as 3, 8, 5. Now, looking at this sorted solution we know that we have to find the extreme solution. So, the solution number 3, solution number 5 are the extreme solution for the given case, and in this case we will be assigning d 3 and d 5 as an infinite value, because we want to preserve the extreme solution of every front.

Now, since the distance crowding distance for 3 and 5 are already calculated, we are left with solution 8; how we can calculate the crowding distance? So, if we look here the solution number 8, the first neighbor is 3 and another neighbor is 5. So, d 8 is equals to d 8 plus. So, f 1 of solution 5 minus f 1 of solution 3 we are taking a positive value divided by f 1 max f 1 min.

So, f 1 5 and f 1 3 we can find it on the second line. Similarly f 1 max and f 1 min also can be identified or calculated from the line number 2. By including the value we will get d 8 is equals to 1, these calculation we perform for the first objective. Now, let us perform the calculation for the second objective as per the algorithm.

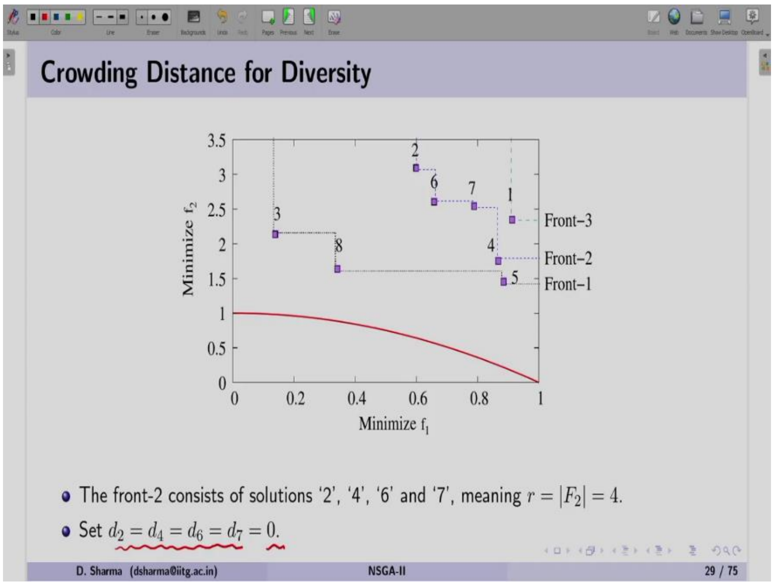
Here the f 2 values of these solutions are given. The second the next step is we have to sort these solution based on f 2, while sorting it we are getting 5, 8 and 3 as a sorted solution. And the extreme solution, now are 5 and 3. Since these are extreme solutions we will be assigning a very large crowding distance value which is infinite as of now.

Now, here you can see for f 1 and f 2 these are the same solution, it is only because we are solving a two objective problem. But, when we are going to solve a 3 objective problem, these solutions can be different. So, the next step is the remaining solution 8 we have to find the crowding distance in the f 2 the formula says that. So, as we look at this sorted solution so, 8 the solution 8 the neighbors are 5 and 3.

So, therefore, d 8 is equals to d 8 f 2 of solution 3 minus f 2 of solution 5 divided by f 2 max minus f 2 min. From the second line we already have the f 2 values of this solution by putting them we get d 8 is equals to 2. The important point which you we can notice here that we have taken the previous component of d 8 and we have taken here.

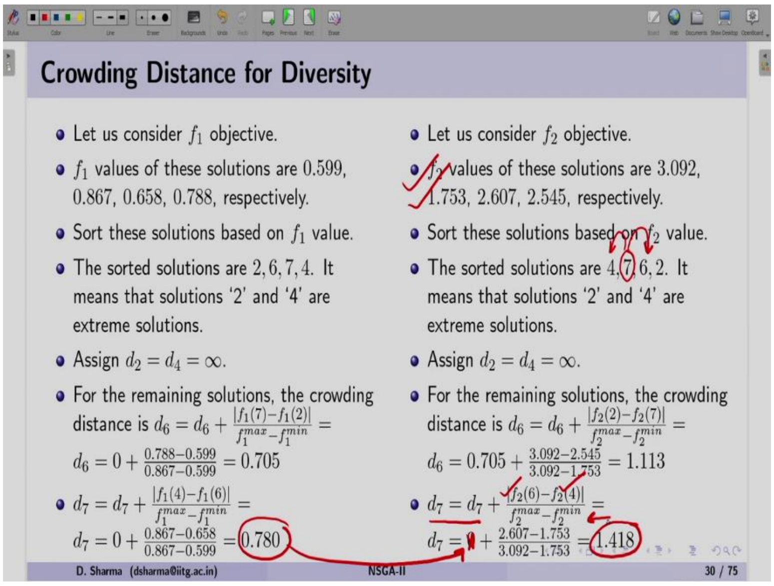
So, the crowding distance of a solution 8 is 2 and d 3 and d 5 is a very large value, theoretically it is infinite. As of now, we have calculated the crowding distance for the front 1, we will be performing the crowding distance for other front as well. It is because as per our discussion this rank and the crowding distance may be useful in the binary tournament selection. So, let us perform the crowding distance for the front 2 solutions.

(Refer Slide Time: 57:05)



Now, looking at the figure we have front 2 now; and in this front 2 solutions 2, 4 and 6 they belong to the so, front number 2. So, the cardinality r says that there are 4 solutions. Next step is we have to assign the crowding distance of each solution equals to 0. So, that is the initialization.

(Refer Slide Time: 57:30)



As per the algorithm let us consider the first objective. In this first objective, we will be writing the objective function values of these solutions. The next step is we have to sort the solution based on f 1. When we are going to sort it, the solutions are sorted in the form of 2, 6, 7 and 4. Looking at this sorted solution, solution number 2 and solution number 4 they are extreme solution and therefore, we will be assigning d 2 and d 4 as a infinite value.

Thereafter, we have to calculate the crowding distance of solutions 6 and 7. Let us take solution number 6 first; for solution numbers 6, the neighbors are 2 and 7; so therefore, the crowding distance of a d 6 is equals to d 6, f 1 of 7 minus f 1 of 2 divided by f 1 max minus f 1 max minus f 1 min. Again, the f 1 7 and f 1 2 we can identify from the line number 2.

Similarly the f 1 max and f 1 min also can be calculated from the line number 2, by putting the value we can get d 6 equals to 0.705. The next solution is the solution numbers 7, the neighbor two solution 7 are 6 and the 4. So, since these are neighbors we are going to find the crowding distance as d 7 equals to d 7 f 1 7 minus f 1 4 minus f 1 6 divided by f 1 max minus f 1 min.

Again the line number two will give us the required value and the d 7 will be calculated as 0.780. Now, we have to consider the another objective f 2 and we have to perform the same calculation. So, the f 2 values of solution 2, 4, 6, 7 are given here, we are going to sort these solution based on f 2 value and the sorted solutions are as 4, 7, 6, 2.

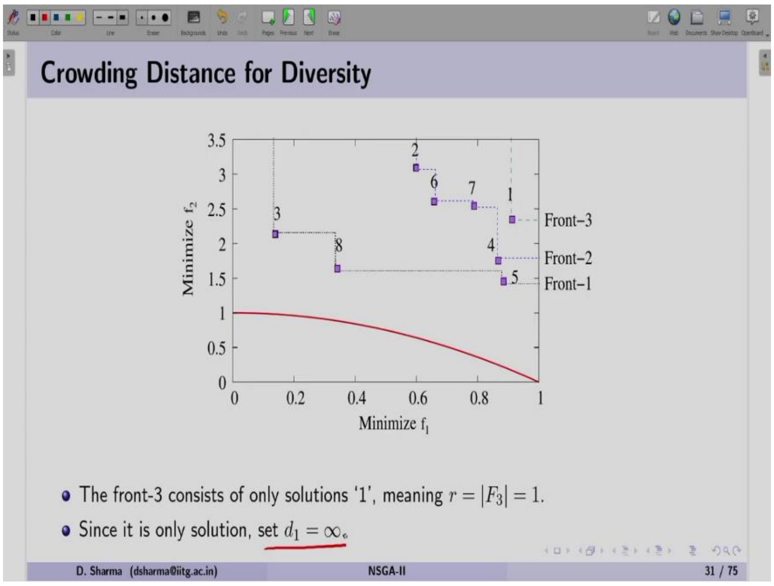
Again solution 4 and 6 are the extreme solutions and therefore, we are assigning the infinite value. So, the same observation since it is a two objective problem the same solutions have the are the extreme solution, but it is not true for the higher number of objectives. Once, we have identified it now we have to calculate the crowding distance of the remaining solution.

So, let us take solution number 6 the neighbors are 7 and 2. So, we will be writing d 6 is equals to d 6, f 2 of 2 f 2 of solution 7 and f 2 max minus f 2 min. All the required values are given on the line number 2 on the right hand side and when are putting the value. So, as you can see that the previous value of a d 6 is taken here, and we calculated the another value and the overall d 6 value is 1.113.

Same thing we are going to perform for solution 7 here, 7 the neighbors are 4 and 6. So, we will be finding d 7 is equals to d 7, f 2 6 minus f 2 4 and then in the denominator we have f 2 max minus f 2 min. Taking all the required values from the second line on the right hand side we will calculate.

So, here this is a typo mistake. So, the same value which we have calculated earlier that shall be taken and then we will get d 7 as 1.141. So, till this step we have calculated the crowding distance for the front 2.

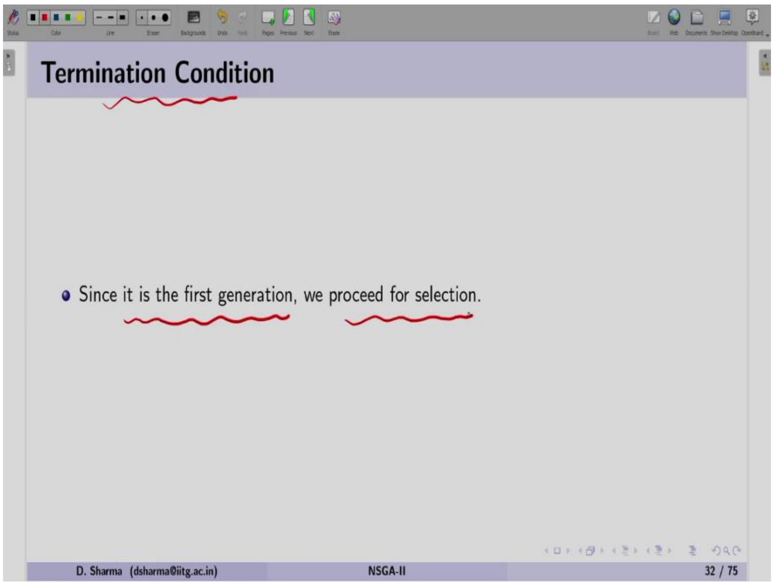
(Refer Slide Time: 62:11)



Now, let us move to the front 3; in front 3 only one solution is lying. Since it is 1 so, we know that it is going to get a infinite value, up to this step we generated the random population. We evaluated the f 1 and f 2 objective values for the all for all the solutions and, thereafter we are assigning the rank and the crowding distance.

So, the rank is was assigned by the dominance depth method or the improved version of it is the non dominated sorting method. And, then we measure the crowding distance, after evaluating and assigning the fitness to each and every solution we are at the decision box.

(Refer Slide Time: 63:01)



In this decision box we look at the termination condition. Since it is the first generation we proceed for the selection. So, as a part 1 of this particular session we have discussed, how we can calculate the rank of its solution and that will that is done using the non dominated sorting and as a part of a diversity we calculated the crowding distance.

So, in this particular session, we have discussed these two main important features of NSGA-II. In the next session, we will be discussing the other features and we will perform the one iteration of NSGA-II followed by the simulations. So, with this the part 1 of this session I am concluding.

Thank you very much.