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

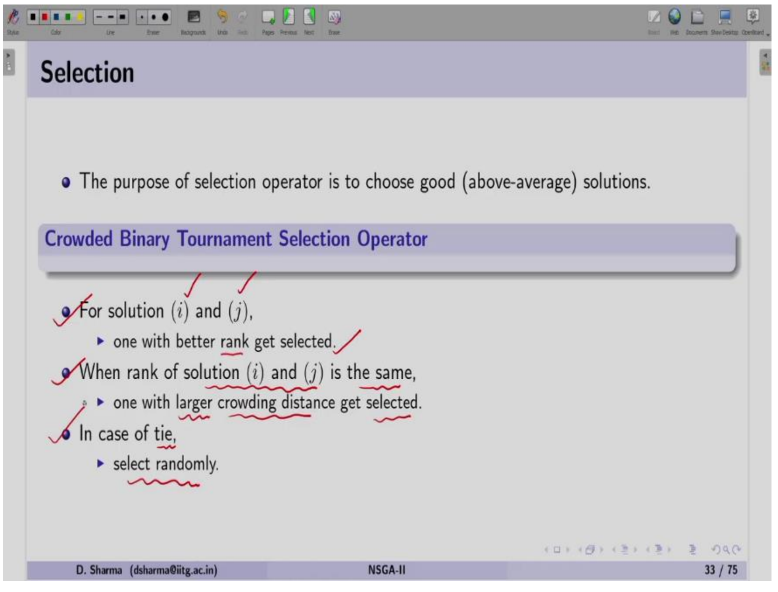
Lecture - 22
Non-Dominated Genetic Algorithm NSGA-II Simulations

Welcome to the part 2 of NSGA II. In the previous session, we have gone through, the we have gone through the introduction about NSGA II. We performed some hand calculations, mainly we perform the non dominated sorting on a set of a solution by taking an example. And afterwards, we also calculated the crowding distance for every front. Thereafter, we have reached to the first step, where we check the termination condition.

Since it was the first generation, so we continue to the selection operator. So, in this particular session, we will start from the selection operator followed by the other operator, that will come inside the loop of generation. Let us begin with the selection operator. So, as we remember, the purpose of the selection operator is to identify good above average solutions.

So, in this scenario, where we have multiple objectives and when we are ranking them; we have rank as well as crowding distance.

(Refer Slide Time: 02:04)



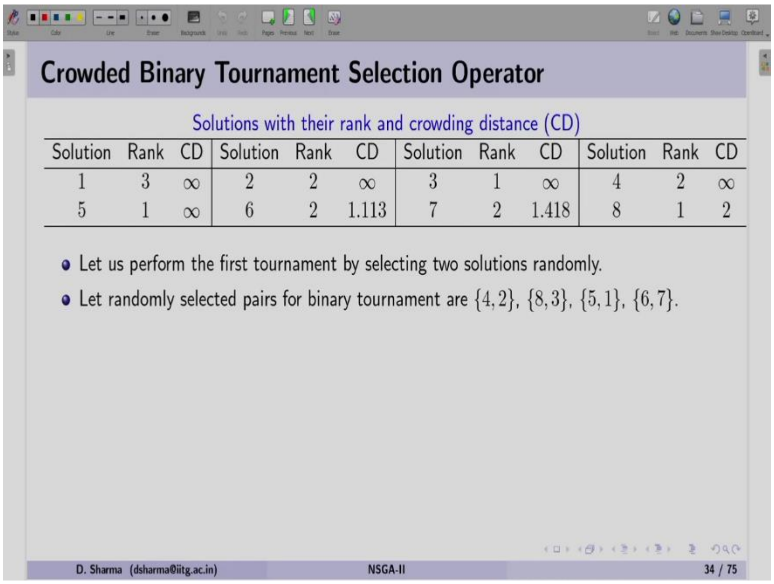
So, NSGA II uses crowded binary tournament selection operator; in tournament selection operator we remember, we discussed with binary coded GA as well as real coded GA. There is a small modification with the crowded binary tournament selection. So, let us see how we can use it for multi objective optimization.

So, as we can see here, inside crowded binary tournament selection operator; we chose two solutions say for example, solution i and solution j and both of them are chosen randomly. Once we chose them, we are going to compare their rank. The solution which is having a better rank get selected. So, in NSGA II the rank 1 solutions are considered better than rank 2 solution. So, if the rank of any of these two solution is smaller, the solution with a better rank get selected.

Second case is, if the rank of these two solution is the same. So, in that case, we will be comparing their crowding distance. The solution which is having the larger crowding distance get selected. And in the third case suppose the rank as well as the crowding distance of solution i and j are same; then we are going to select one solution randomly.

So, here looking at these three condition, it can be these conditions can be implemented using if, else if, and else condition. And we can use the binary tournament selection for the with the NSGA II.

(Refer Slide Time: 04:02)



So, let us take the same solutions that for which we have calculated the rank. So, as we can see in the table on the top, the solution is given, rank is given and CD means the crowding distance. So, for every solution, we have shown the rank and the crowding distance.

So, as we remember that in the tournament selection, we will be selecting two solutions randomly. Let us take this as our first tournament. Let the randomly selected pair for binary tournament selection are 2, 4; 8, 3; 5, 1, and 6, 7; meaning that, we are going to compare 2 and a 4 first and followed by 8 and a 3 and the other solution as given in the pair.

So, let us compare one by one now. If we compare solution number 2 and a 4. So, solution number 2 and 4; if we look their rank, they both of them have a rank 2. So, meaning that they are, they have the same rank. And if we look at the crowding distance that, that comes out to be infinite infinity; meaning that these solutions are the extreme solution in front 2.

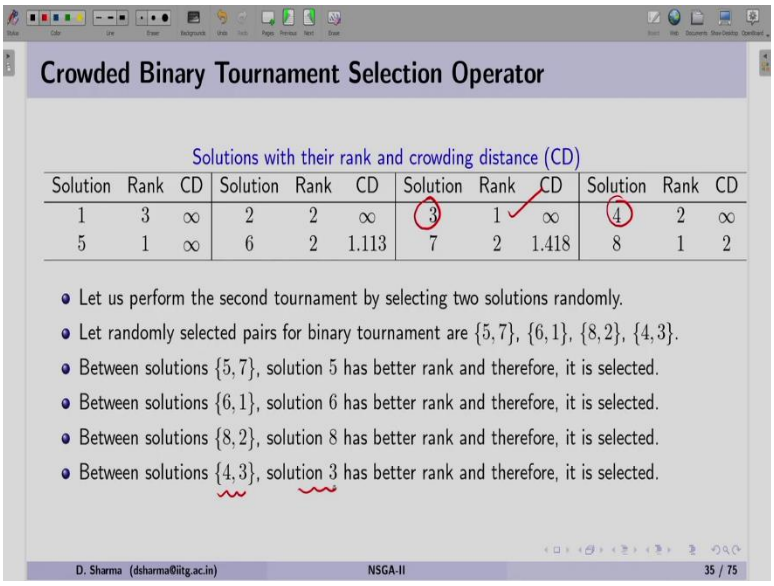
In this scenario, since we have same rank and same crowding distance; we are going to choose one solution randomly. Let us assume that we select solution number 4. The another pair of solution for performing binary tournament is 8 and 3. If we compare solution 8 and 3, now we can see that the rank of these two solutions is the same; but if we look at the crowding distance of these two solution now, here we can see that solution 3 is having more crowding distance value than solution 8.

Meaning that, we are going to select solution 3. The another pair which we selected randomly is 5 and a 1. So, solution 5 and a 1 can be seen here. Looking at their rank we can see that the solution 5 will be selected, because it has a better rank. Finally, the two solution which are left out are 6 and a 7.

If we compare solution number 6 and 7, so the rank of these two solution is the same. But we look at the crowding distance value, so we can see that the solution 7 has more crowding distance and therefore, the solution 7 is selected.

We can see that since it is the first tournament selection, so we selected solution number 4, 3, then 5 and finally, the 7. So, basically we selected four solution.

(Refer Slide Time: 07:37)



So, now as we remember, we are going to perform the tournament selection one more time, so that the selected solution will become equal to n, which is the size of the population. So, this is the second tournament which we are going to perform. Let us assume that the we have picked randomly 5 and a 7 in a pair, then 6 and a 1, 8 and a 2 and 4 and a 3. So, let us compare these solutions one by one.

So, when there is a tournament between solution number 5 and 7; so we can see 5 and a 7, solution 5 has a better rank. So the solution this solution will be selected. Thereafter, we have 6 and a 1; so looking at solution number 6 and a 1, we can see solution 6 has a better rank, therefore it is selected.

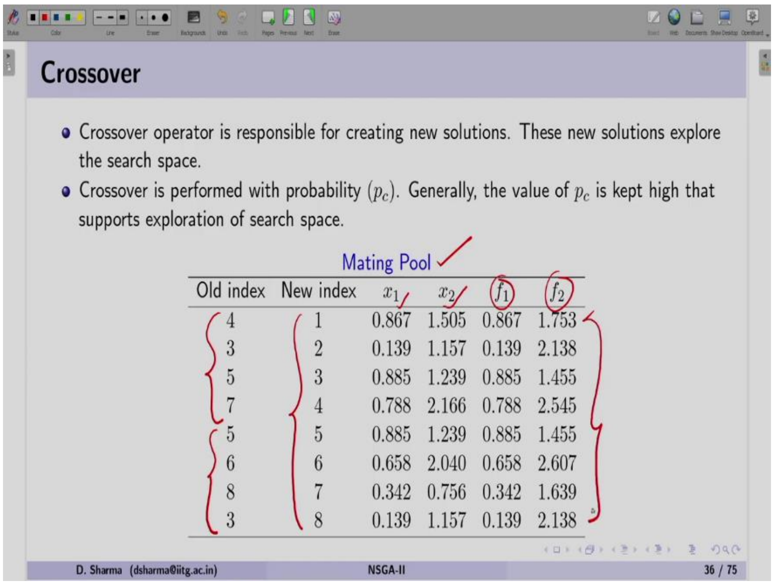
Now, randomly picked solution 8 and a 2 when we compare, so solution 8 and solution number 2. Now, we can see solution number 8 has a better rank. So, therefore, this solution is selected. Finally the last pair is 4 and a 3. When we are going to compare them, we can see the solution 3 has a better rank than solution 4.

So, in this case, we will be selecting the solution 3. So, from our discussion we can see that, whenever we are comparing two randomly chosen solution, first we look at their rank; if the rank is different, the solution which is having a better rank will be selected.

However, if the these two solutions have the same rank, then we look into the crowding distance. And after comparing it, the solution which is having a larger crowding distance is selected. And in a very rare condition if rank and crowding distance both are same, then we are going to select any of the solution randomly.

So, we remember that as per our algorithm, the step inside the loop of generation; we started with the selection and thereafter comes variation. So, the first operator which we apply under variation is crossover operator.

(Refer Slide Time: 10:16)

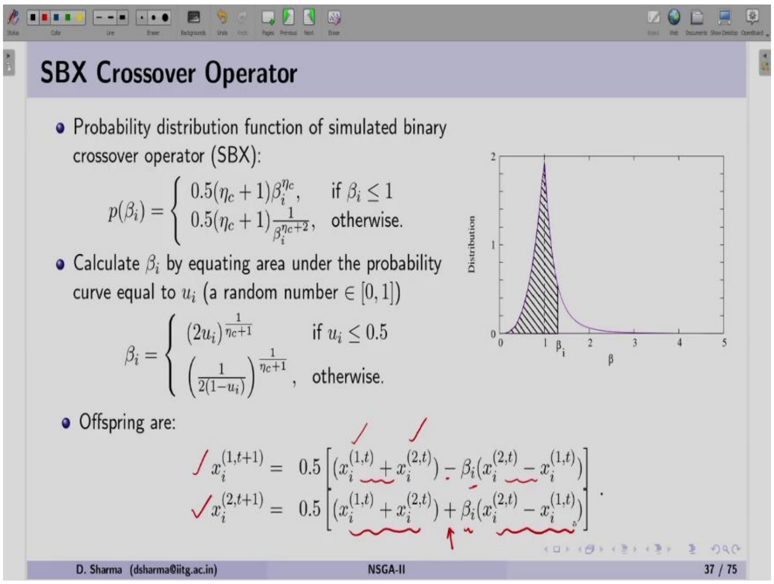


So, let us discuss this crossover operator now. Now, as we know that the crossover operator is responsible for creating new solutions. These new solutions explore the search space. We generally perform crossover with a probability called p c. So, the crossover probability and we kept the value of a p c high, so that it can support exploration of the search space.

From the binary tournament selection or we can say crowded binary tournament selection with NSGA II. We selected first four solution as we can see on the on the first column of the table. So, these are the solutions selected from the first tournament; thereafter we selected another four solutions in the second tournament.

Now, the same solutions which are now copied into the mating pool, we are giving a new index as given in the second column of the table. There x 1 x 2 value and as well as the f 1, f 2 values are also shown. It is important to note that, we do not need f 1 and f 2 as of now; because the crossover is performed with the decision vector. So, therefore, f 1, f 2 has no role; but we are showing as a representation purpose.

(Refer Slide Time: 11:58)



$$p(\beta_i) = \begin{cases} 0.5(\eta_c + 1)\beta_i^{\eta_c}, & if\ \beta_i \le 1 \\ 0.5(\eta_c + 1)\dfrac{1}{\beta_i^{\eta_c+2}}, & otherwise \end{cases}$$

$$\beta_i = \begin{cases} (2u_i)^{\frac{1}{\eta_c+1}} & if\ u_i \le 0.5 \\ \left(\dfrac{1}{2(1-u_i)}\right)^{\frac{1}{\eta_c+1}} & otherwise \end{cases}$$

$$x_i^{(1,t+1)} = 0.5\left[\left(x_i^{(1,t)} + x_i^{(2,t)}\right) - \beta_i\left(x_i^{(2,t)} - x_i^{(1,t)}\right)\right]$$

$$x_i^{(2,t+1)} = 0.5\left[\left(x_i^{(1,t)} + x_i^{(2,t)}\right) + \beta_i\left(x_i^{(2,t)} - x_i^{(1,t)}\right)\right]$$

NSGA II 2 uses SBX cross over operator; this crossover operator we have understood while going through the real coded GA. As a recap of simulated binary crossover which is known as SBX cross over operator, let us see how it works. So, the probability distribution function for SBX operator is given here as we can see.

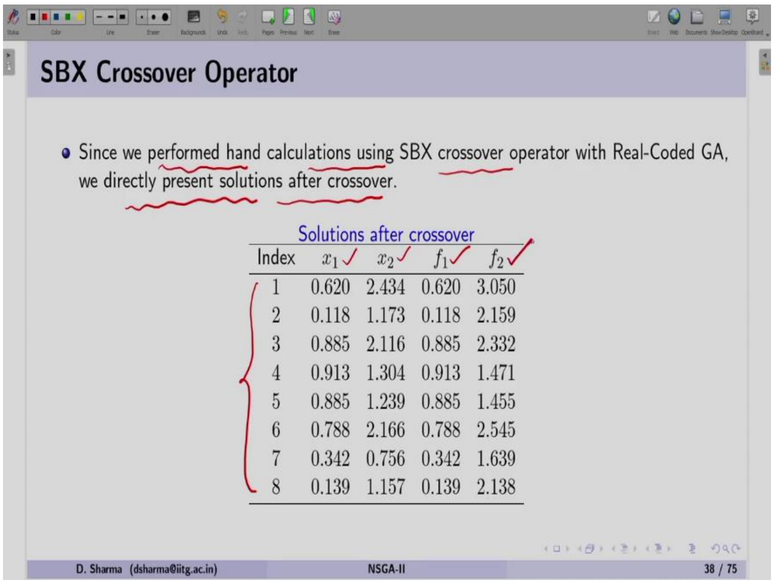
Now, this is a non-linear probability function, which can be seen in the figure on the right hand side. In order to calculate this beta, we calculate by equating the area under the probability curve to u i, and u i is a random number between u 0 to 1. As we can see on the figure on the right hand side that, when we are equating the area and try to find out what is beta i.

After simplification based on the u i value, which is a random number, that will be generated by the computer for us; if the random number is smaller than or equal to 0.5 we will be using the formula on the top and otherwise we will be using the another formula.

We remember that, this eta c has a role here; because that will tell about the spread of the curve. Using the value of a beta i, we are taking the average of the two parents. So, as we understood that, we selected parent 1, parent 2 and we find the average of it minus times of beta of the difference between the two parents.

Similarly, the second, second offspring will be generated by taking the average; but now we are adding this beta i and the difference of these two parents solution.

(Refer Slide Time: 14:11)



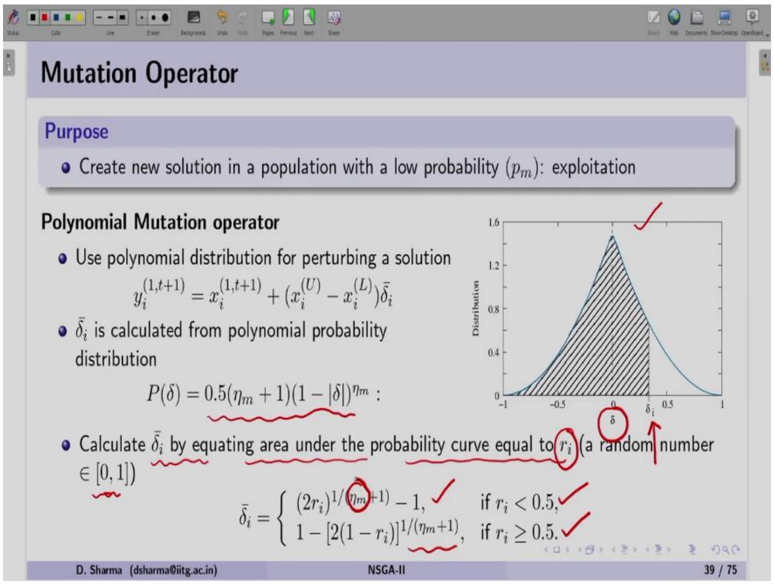
Solutions after crossover

Index	x_1	x_2	f_1	f_2
1	0.620	2.434	0.620	3.050
2	0.118	1.173	0.118	2.159
3	0.885	2.116	0.885	2.332
4	0.913	1.304	0.913	1.471
5	0.885	1.239	0.885	1.455
6	0.788	2.166	0.788	2.545
7	0.342	0.756	0.342	1.639
8	0.139	1.157	0.139	2.138

Since, we perform the hand calculation using SBX operator with real coded genetic algorithm; we directly present solution after crossover. So, the table which we can see here, these are the solutions; so basically x 1 and x 2 value, the modified values after crossover we can see here. Just for the representation purpose, we are showing the f 1 and f 2 values.

Once we perform the crossover operator, under the variation we perform mutation. So, the mutation as we remember, the purpose of the mutation is exploiting the search space by perturbing the solution. So, let us have a recap of the mutation and then we will follow our hand calculation.

(Refer Slide Time: 15:15)



$$y_i^{(1,t+1)} = x_i^{(1,t+1)} + (x_i^{(U)} - x_i^{(L)})\bar{\delta}_i$$

So, the purpose of this mutation operator is to create new solution in a population with the row with the low probability p m. So, we remember that, this probability p m is known as the probability of mutation and which we generally kept as probability of mutation is 1 divided by n, n is the number of variable. So, it is a thumb rule.

$$P(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{\eta_m} :$$

And this mutation will be helpful for exploitation. NSGA II uses polynomial mutation operator; we remember that if the solution which is generated by the crossover operator, that is added with the delta i which is multiplied with the upper and lower bound of the variable. And this will give me the mutated solution using polynomial mutation operator.

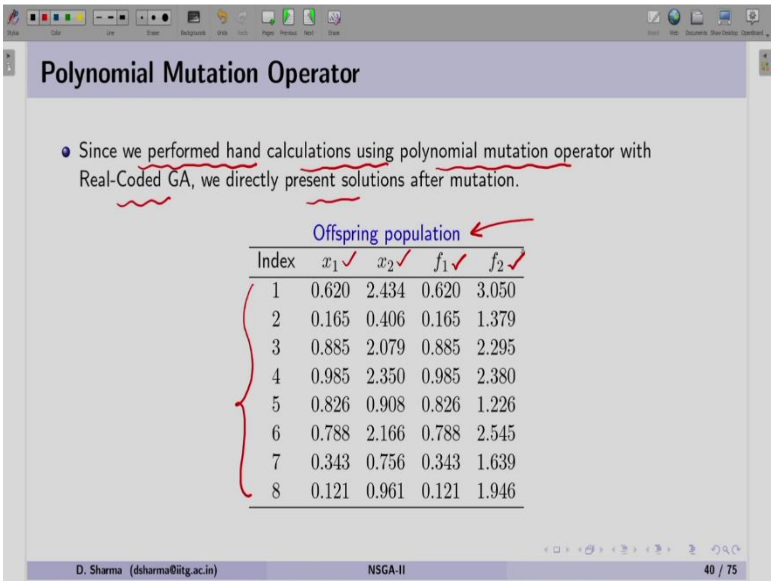
$$\bar{\delta}_i = \begin{cases} (2r_i)^{\frac{1}{\eta_m+1}} - 1, & if\ r_i < 0.5 \\ 1 - [2(1-r_i)]^{\frac{1}{\eta_m+1}}, & if\ r_i \geq 0.5 \end{cases}$$

The probability function for polynomial mutation is again shown here, which is a non-linear function. Now, in this case as we can see on the right hand side, this particular function, probability distribution function we are going to use to calculate the value of delta i.

Similar to the previous SBX operator, we are going to calculate delta i by equating the area under the probability curve equals to r i. And r i is again a random number between 0 and 1. So, basically when we are equating here. So, the area under the curve we are equating with the random number.

Since this random number will be generated by a computer. So, we are going to get two values of or two equations of delta i. If r i is smaller than 0.5, then we will be using the formula given on the top. If r i is greater than an equals to 0.5, then we will be using the another formula. Here as per our discussion earlier, eta m is a user defined parameter that we set and that also takes cares of the spread of this probability distribution curve.

(Refer Slide Time: 17:54)

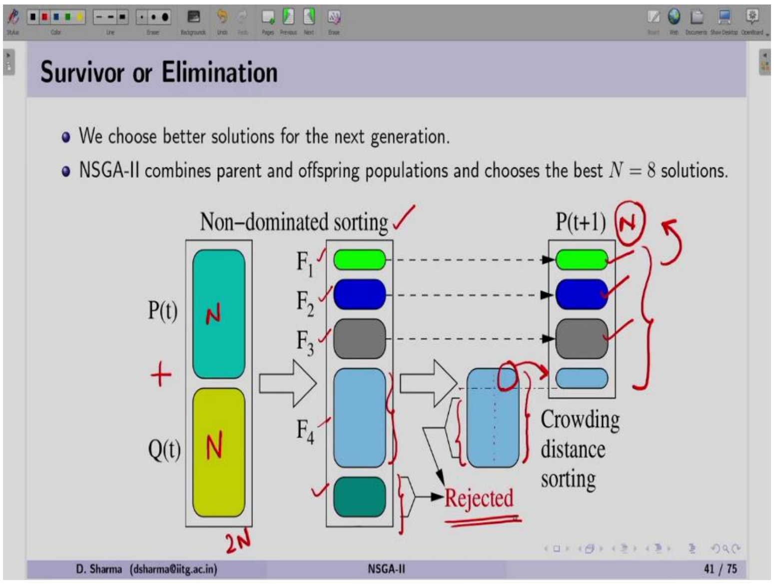


Index	x_1	x_2	f_1	f_2
1	0.620	2.434	0.620	3.050
2	0.165	0.406	0.165	1.379
3	0.885	2.079	0.885	2.295
4	0.985	2.350	0.985	2.380
5	0.826	0.908	0.826	1.226
6	0.788	2.166	0.788	2.545
7	0.343	0.756	0.343	1.639
8	0.121	0.961	0.121	1.946

Since we performed hand calculation using polynomial mutation with real coded GA, we directly present our solution. So, here we are referring this particular population as offspring population and the solution with the modified value of x 1 and x 2 are given in the table.

Since we perform the mutation at a low rate, so we will see that some variable will be modified and some variable will remain the same. As a representation purpose, f 1, f 2 values are shown.

(Refer Slide Time: 18:41)



Now, once the offspring population is generated; at this particular stage, we have parent population as well as the offspring population. So, NSGA II has come up with the survival or elimination stage, where it takes the front or the solution with their rank as well as crowding distance.

So, let us understand how this survival stage is performed with NSGA II. So, here the purpose of this stage is, we choose better solution for the next generation. NSGA II combines parent and offspring population and among the set, we are going to choose the best N solution and we remember that N is the population size.

For the current example, N is equals to 8. If we see how the survival or the elimination is performed; so we can see we have parent population which is of size N, similarly we have offspring population which is of size N.

So, basically we have a size of 2 N and the task is we have to select the N best solution under P t plus 1. So, what we do here is, we combine them and on this combined population of P plus Q, we perform non dominated sorting. As graphically we can understand that, some solution will be lying in F 1, similarly in F 2, F 3 and the other fronts as well.

Thereafter, we will be selecting the solution one by one. Say here, as we can see that the size of F 1 is small, it can be accommodated within N. Similarly, we can copy the solutions of F 2 and F 3 as well. So, these three front solutions, we can very well accommodate into P t plus 1 of size capital N. Now, since still there is a space available to make the size N, we are going to include the front 4 now.

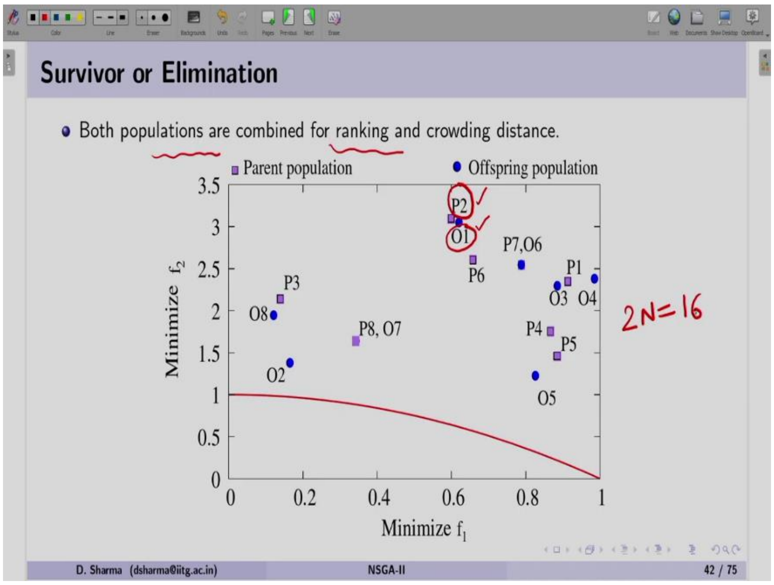
Now, as we look, there are many solutions which are lying in a front 4 and when we want to copy, we cannot copy all of them. So, in this particular case, we since we have already calculated the crowding distance. So, based on the crowding distance, the solutions will be sorted in a descending order. So, the solutions which are coming on the top; they will be selected here, so that all these solutions will constitute population size of N.

Now, in this particular elimination; we can see that below this particular line, all these solutions and as well as the front which are worse than F 4 all of them will be rejected. So, meaning that, if we are adding P plus Q, so we have 2 N population. So, combinedly let us find their rank, let them let all these solution sorted in different fronts and thereafter we are going to calculate the crowding distance.

So, we will copy front by front these solution into the new population, which is called P t plus 1. So, as we can understand from the graphical example here, we will be copying the front 1. Since the size of front 1 is small, means the number of solution is less than the size of capital N, we will copy them, thereafter F 2 and similarly F 3.

Once the solution from F 4 are going to be selected; since the number is large, so we selected using the crowding distance, the solution having more crowding distance will be selected. And that is why we sorted the solution in front 4 based on the larger crowding distance, basically in a descending order.

(Refer Slide Time: 23:36)

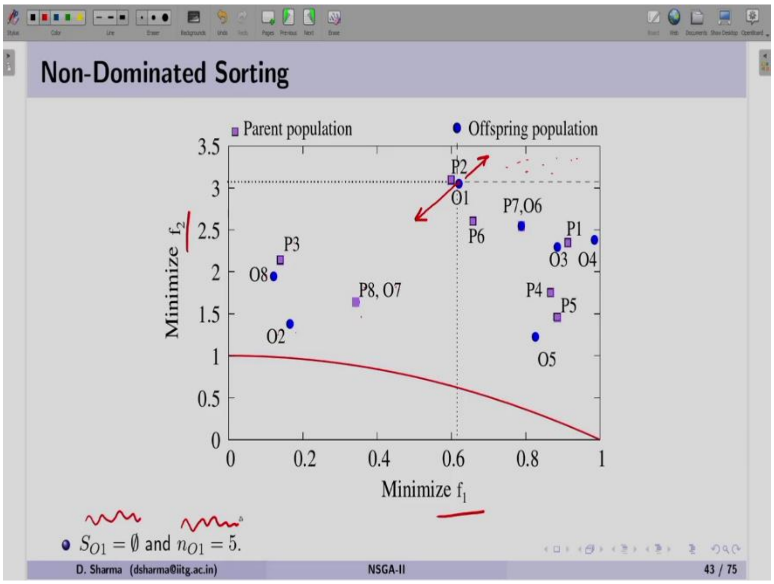


Now, let us take the same solutions that are generated after crossover and a mutation that is offspring and the initial population which we consider as a parent population for the given example. As we can see here now, both the population are combined for ranking and crowding distance. Now, here the representation says that, if we are writing for example, P 2; this means this is the second solution of parent population.

Similarly, if we look solution say O 1; this means, it is the offspring solution and that is the first solution in the offspring population. This particular notation we are using, so that we can differentiate the parent population and offspring population for understanding how this elimination works.

Now, in this case as you can see that, we have 2 N solution. So, basically we have 16 solutions here. Now, we are going to rank them and then find the crowding distance.

(Refer Slide Time: 24:57)



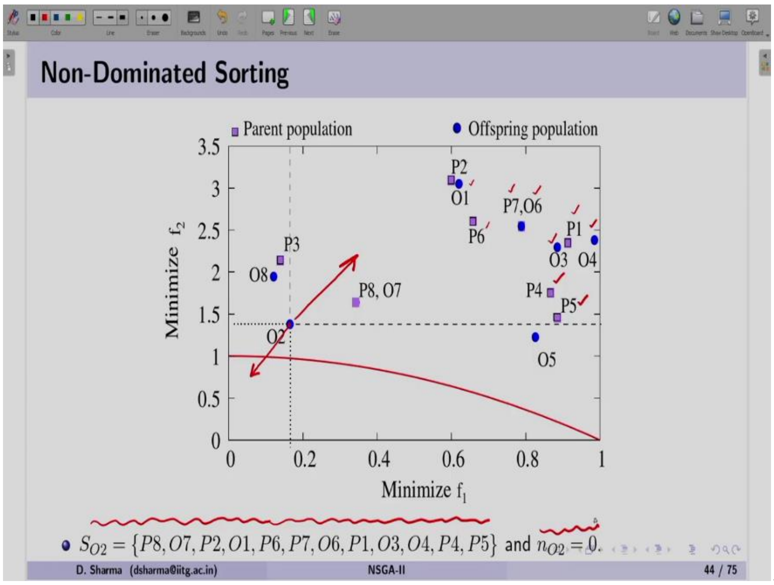
Now, since this problem is minimization of f 1 and minimization of f 2. As per our earlier discussion, if we take for example, solution number O 1, means offspring 1. If we look, if we consider this as a reference solution and looking into the first quadrant of solution O 1; if there is any solution means that, those solutions will be dominated by O 1.

Now, looking at the present case, since there is no solution in the first quadrant of O 1; so as we can see that S O 1 is empty set. Similarly, if we take the reference O 1 and look into the third quadrant, so these are the solutions such as P 3, O 8, O 2, P 8 and O 7; these solutions will be dominating solution O 1.

So, if we count them, it is 1, 2, 3, 4 and 5 solution and therefore, n O 1 is 5. To understand the rank of the or the non-dominated sorting of these solutions. We are using the same notation of the non-dominated sorting; that is the set S p and we also calculate n p, S p refers to the set in which the solutions are dominated by the solution P.

Similarly, the counter n Q refers to the number which says that, how many solutions are dominating solution P. So, we are following exactly the same notation here and we will take, we are calculating the S set as well as n counter for every solution.

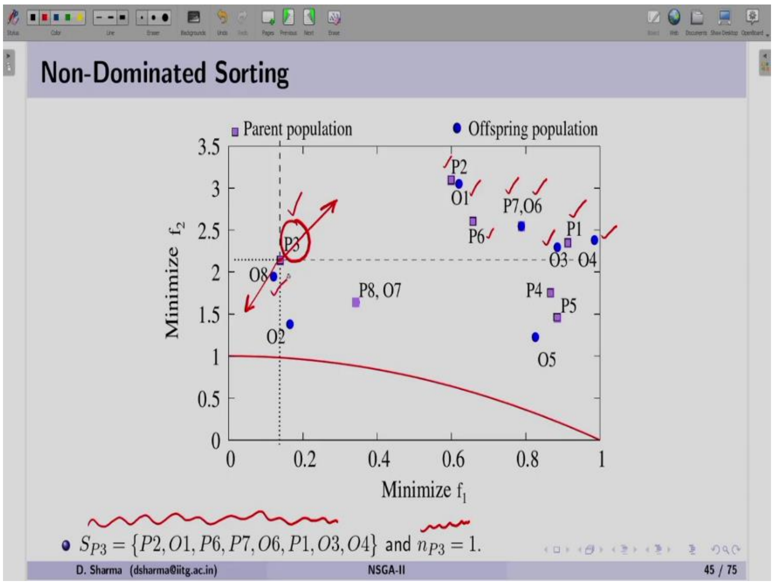
(Refer Slide Time: 27:06)



Now, let us take the solution number O 2. Now, when we are taking this as a reference and looking into the first quadrant of O 2; we can say that solution P 2, O 1, P 6, P 7, O 6, P 1, O 3, O 4, P 4, and P 5.

All these solutions are dominated by solution O 2. And therefore, at the bottom we can see, the set S O 2 is comprising of all these solution. Taking the reference of O 2 and looking into the third quadrant. We can see n O 2 and this n O 2 is 0; because there is no solution which is dominating O 2.

(Refer Slide Time: 27:59)

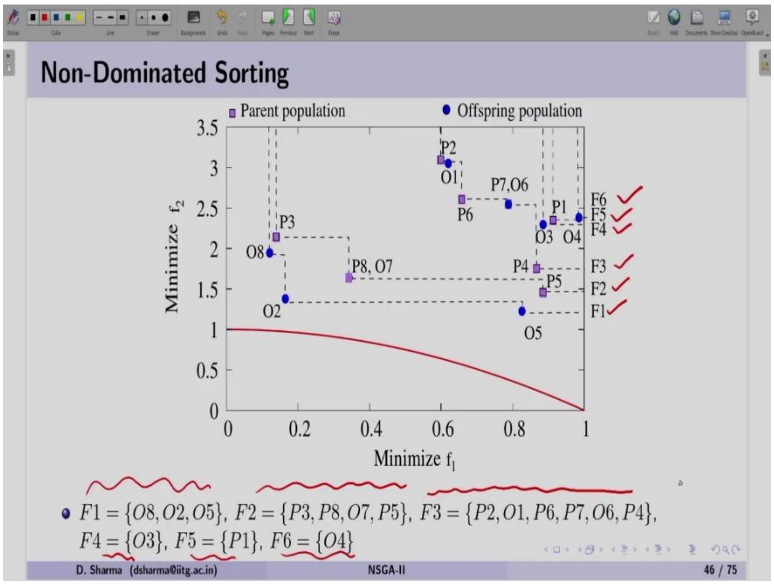


If we take the another case to understand about the ranking, the non-dominated sorting. So, here we have solutions say P 3. So, we have taken this solution P 3; it is only because just to see that, when we are taking the parent solution, there the solutions, the number of solution which it will be dominating or it will be dominated by that will change.

Looking at P 3 now; in the first quadrant of P 3, we can see there are many solutions such as P 2, O 1, P 6, P 7, O 6, P 1, O 3, and O 4, all these solutions will be dominated by P 3. So, at the bottom we can see that S P 3 is given. Similarly, if we look at the third quadrant, only solution O 8 is going to dominate P 3 and therefore, n P 3 is 1.

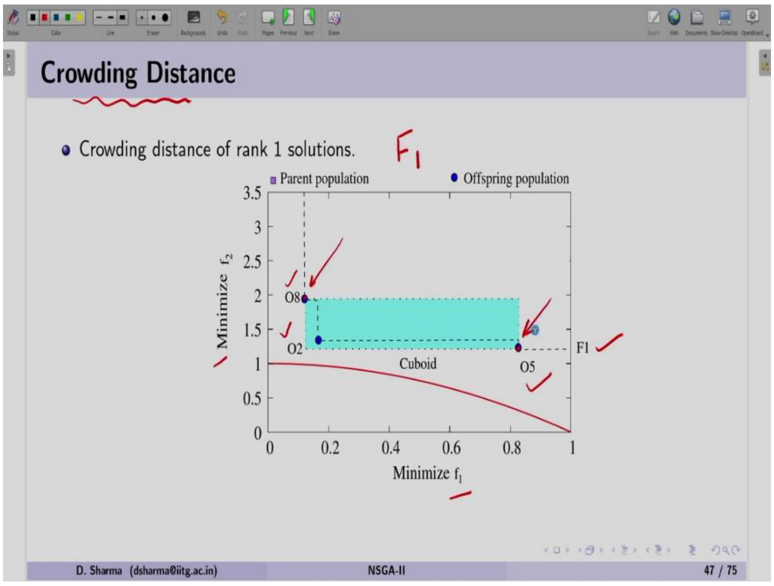
So, the observation is that, the solution P 3 when we perform the ranking at the beginning to evaluate the rank as well as the crowding distance, solution P 3 was the non-dominated solution. But after performing crossover and a mutation; we have better solutions such as solution O 8, which is now dominating P 3.

(Refer Slide Time: 29:39)



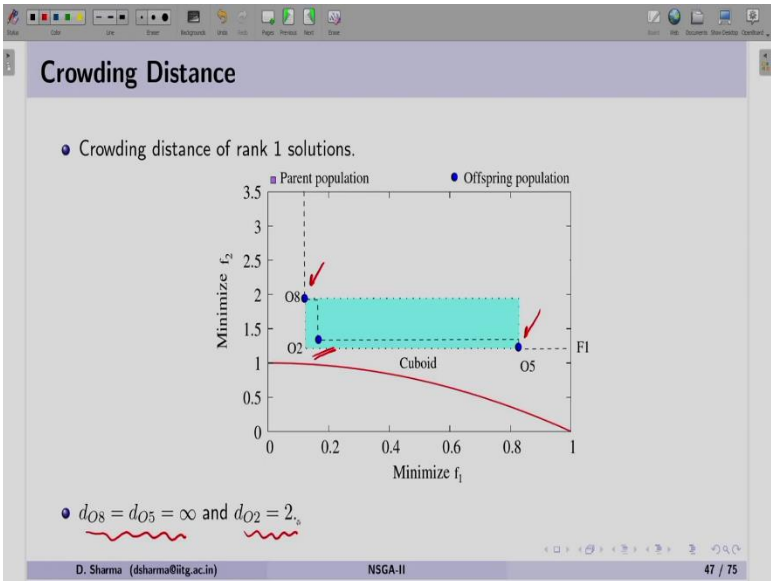
By following the similar procedure of finding the S P set and n P and following the procedure as we have discussed here. We can see the 16 solutions are now sorted into six fronts from F 1 to F 6. By following those thing we can see the front 1 consist of O 8, O 2 and O 5; similarly front 2 consist of P 3, P 8, O 7 and P 5; and similarly the other front is given at the bottom of the slide.

(Refer Slide Time: 30:33)



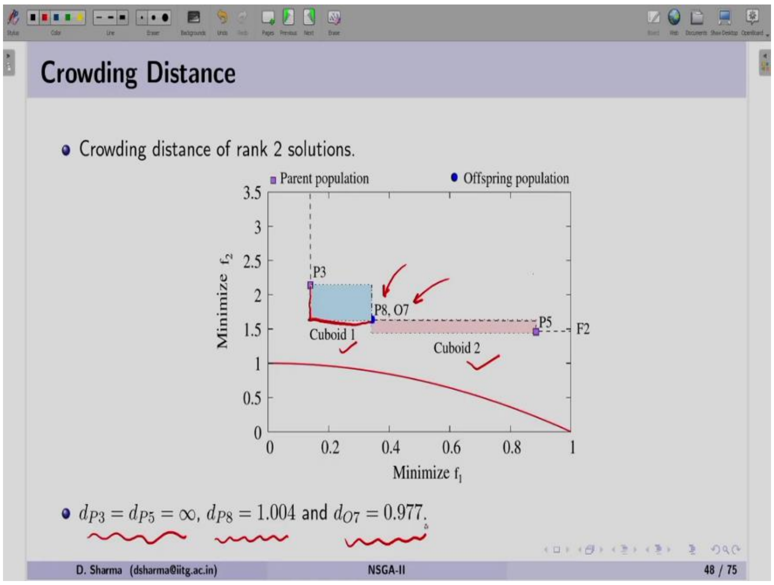
Now, once we have calculated the rank of all the solution; now we have to calculate the crowding distance of each solution. We will begin the crowding distance of the solution which are lying in front 1, basically the rank 1 solution. So, the front 1 solution r given in the figure so O 8, O 2 and O 5 are here. Since it is a two objective problem we can find it out that, the solution number 8 and solution number 5 both are extreme solutions.

(Refer Slide Time: 31:14)



Since these are extreme solutions, so we are going to give a crowding distance value as a infinite value. Now, we have to just calculate the crowding distance of a remaining solution, which is O 2. Since it is only the one solution, we know that the crowding distance is going to be 2 for the given solution.

(Refer Slide Time: 31:40)

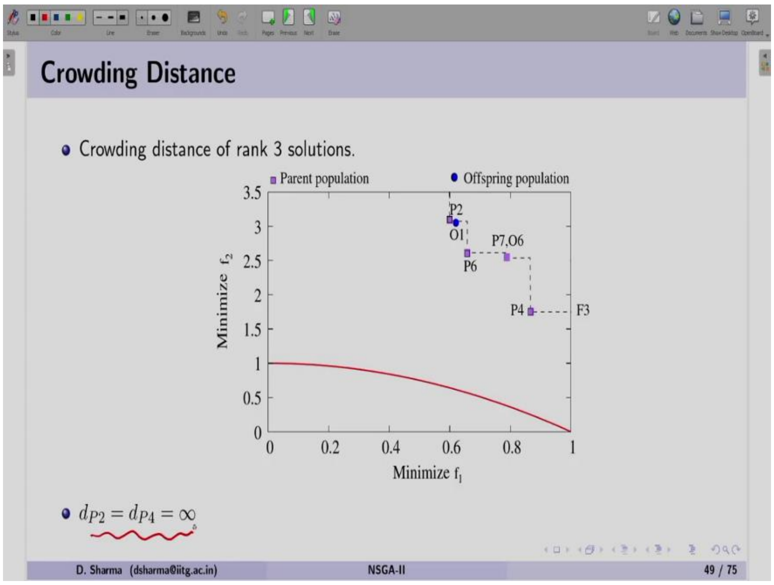


Similar to the crowding distance of front 1; now let us calculate the crowding distance of solution having the rank 2, means the front 2. In the figure, we can see the solutions are we have in this front are P 3, P 8, O 7 and P 5. In this particular case, we can find that the solution P 3 and the solution P 5 they are extreme solution. So, we are going to give them in a infinite value as a crowding distance.

Now, by performing the calculation for say cuboid 1 and cuboid 2; so as we remember our crowding distance that says that, the that says that the crowding distance basically tells about the perimeter.

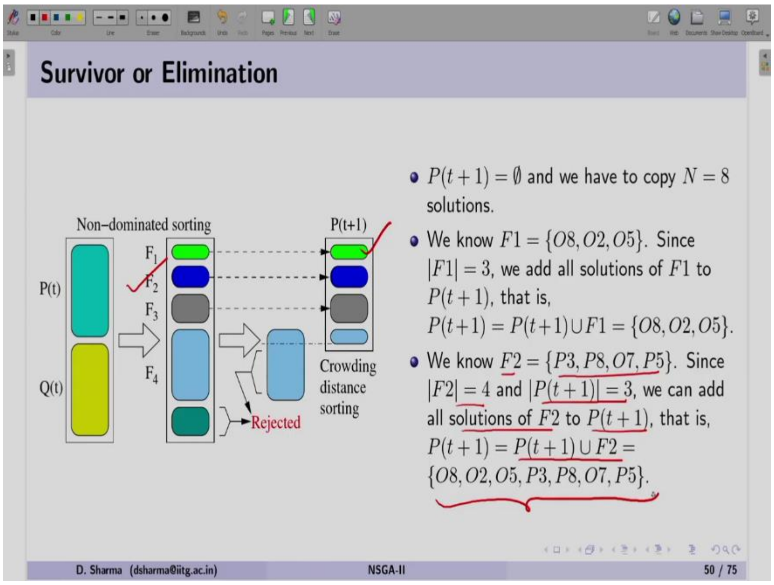
So, in this case the perimeter of a cube 1 will become crowding distance of solution say P 8 and the perimeter of cuboid 2 will become the will become the crowding distance of O 7. After calculating the crowding distance, we get the crowding distance of P 8 is 1.004 and crowing distance of 7 is 0.977.

(Refer Slide Time: 33:07)



Similarly, we are going to calculate the crowding distance for the front 3 solutions. Now, in this case looking at this front; solution number P 2 and solution number P 4 they are extreme solutions, so they are going to get infinite crowding distance value theoretically. We are assuming that we can calculate the crowding distance of the other solution that are lying in the front 3 as well as in the front other fronts.

(Refer Slide Time: 33:42)

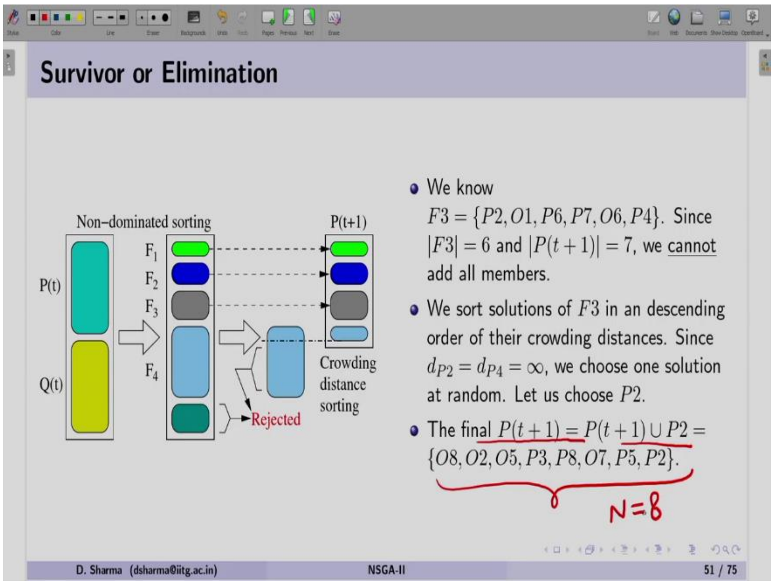


Now, let us apply the survival or the elimination of NSGA II. Looking at the figure, we set P t plus 1 as a empty set and we have to copy N equals to 8; because the size of the population is 8.

Now, let us take front 1. So, as per the figure, we will take front 1 now. Now, in this particular front we have three solutions. Now, since the remaining size in P t plus 1 is 8; so we can accommodate all the solution of F 1 in P t plus 1, meaning that P t plus 1 is equals to P t plus 1 union F 1. So, these three solution are copied into the next generation population.

Once it is done, now let us consider the front 2; because front 1 is already copied. Looking at front 2 now, it has 4 solutions and the size of P t plus 1 is 3; meaning that 4 plus 3 means 7. So, we can accommodate all the solution of F t F 2 into P t plus 1. So, therefore, we are adding the solution of P 2; we are adding the solution of F 2 into P t plus 1. So, as we can see the P t plus 1 is now accommodating 7 solutions.

(Refer Slide Time: 35:14)



Now, since the size of P t plus 1 is 7, it can accommodate one more solution. Now, looking at the F 3 which is currently consisting of 6 solution; so the remaining size is 1, the available solutions are 6. So, we cannot accommodate all of the solution. As per the suggestions we are, we have to sort our front as per the crowding distance in a descending order and then we have to choose the solution. So, we sort all the solution of F 3 in an in a descending order of their crowding distance; since d P 2 and d P 4 is equal to infinity, it is only because the solutions were the extreme solutions.

So, we have to copy one of them. So, we choose one solution at random, let us take P 2. So, in this case, or in this scenario, the final P t plus 1 equals to P t plus 1 union P 2; meaning that we have added the P 2 solution now and it consist of 8 N equals to 8 solution.

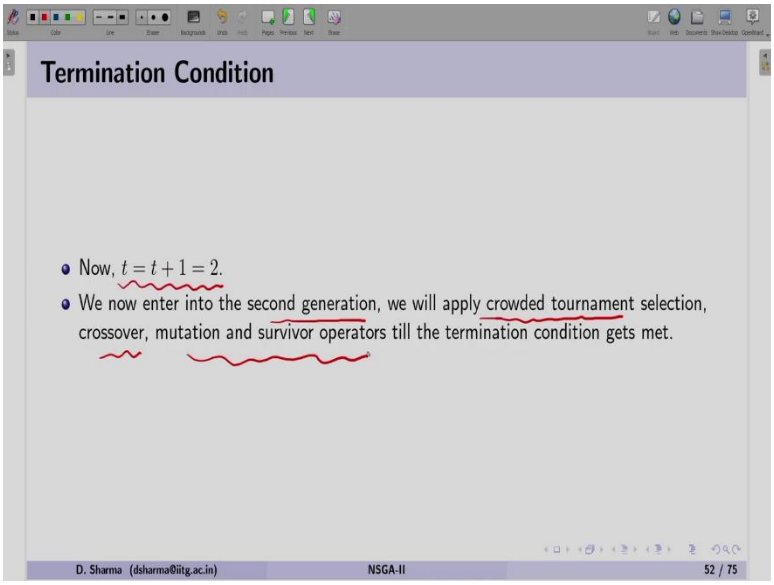
So, from this particular elimination we can see that, when we are adding the parent population and offspring population and combined combinedly doing the ranking as well as the crowding distance calculation. This allows us to choose the best solution or we can say good solution.

When we are adding and selecting good solution; in this case, we will be always having good solution and we are not going to miss any of the any good solution from the, either from the parent population or from the offspring population.

In the scenario, when suppose we run NSGA II for a longer time and solving the same problem; we can get a scenario when the front one itself consists of 16 solution. So, in this case when the front 1 can have all the solution and there is no front 2, front 3, front 4; then all these solutions are going to have the same rank. And this front 1 cannot be accommodated into P t plus 1.

So, the solutions which we are going to select is, we will sort all the solution in front 1 based on the crowding distance. And then after sorting them in a descending order, we are going to choose the top N solution for the next generation population. And this is the way, we will be selecting good solution generation by generation.

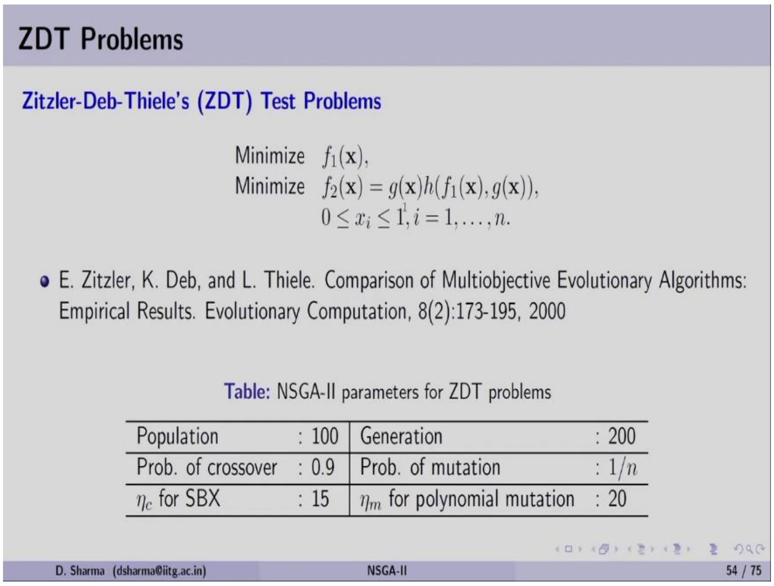
(Refer Slide Time: 38:28)



So, this is the last step in the NSGA II and thereafter we increase the counter. As we can see here now, t is equals to t plus 1 equals to 2; so this means it is going to be the second generation. Generally, we allow many number of a generation, so that the algorithm should converge to the pareto front. So, inside this loop in the second generation, we will again be applying the crowding distance tournament followed by crossover, mutation and the survivor operators till the termination condition get satisfied.

We now start the simulation of NSGA II; the first set of problems which we have taken is from the ZDT set.

(Refer Slide Time: 39:18)



$$Minimize \quad f_1(x)$$

$$f_2(x) = g(x)h\big(f_1(x), g(x)\big)$$

$$0 \leq x_i \leq q\ 1, i = 1, \ldots, n.$$

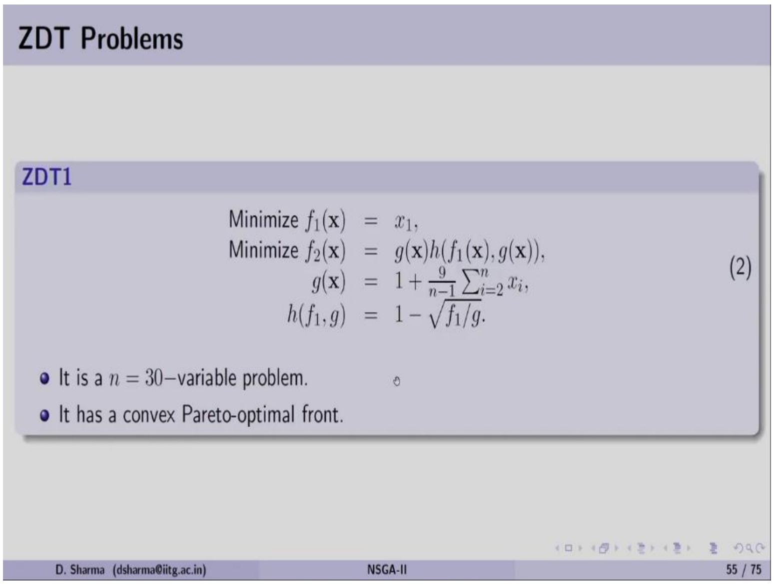
Now, as we can see here ZDT is stands for the Zitzler, Deb, Thiele; these are the author of these problems, so their surnames are used as ZDT test problems. These ZDT problems are two objective problem, where we want to minimize f 1 and minimize f 2 which is composed of functions say g and h.

All the variables are lying between 0 to 1 and we can see that all these test problems can be found on the paper given here. The ZDT problems are considered as the benchmark problems; because when we develop any multi objective evolutionary algorithm, we want to test our algorithm whether the algorithm or EC technique is working fine or not.

Second thing is we can see that these ZDT problems they do not have any constraints, they have only the variable bounds. Therefore these problems are also called as box constrained optimization problem. For solving all these ZDT problems, we have taken NSGA II parameters as the population size is 100.

The generation is 200, probability of crossover is 0.9, probability of mutation is 1 by n; this is the thumb rule we are following. For s b SBX operator eta c is 15, for polynomial mutation the eta m is 20.

(Refer Slide Time: 41:10)



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

$$Minimize \quad f_2(x) = g(x)h(f_1(x), g(x))$$

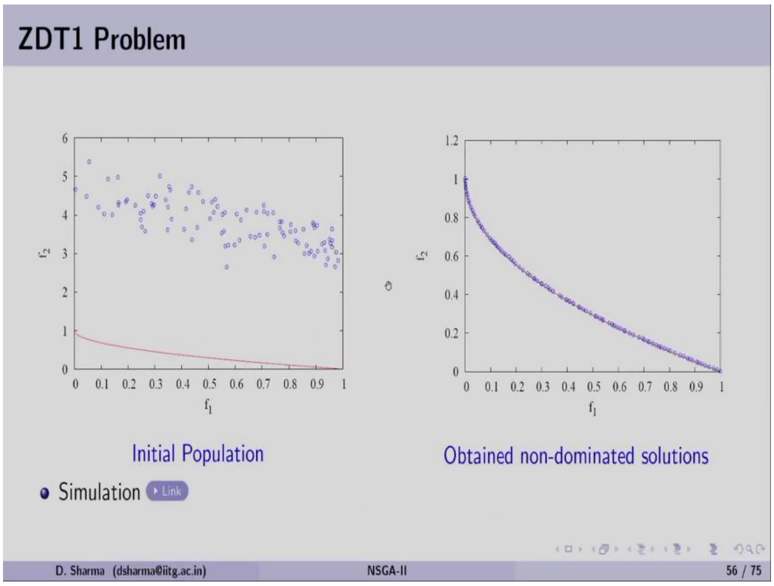
$$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$$

$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}$$

So, let us begin our first problem that is ZDT 1 problem. As we can see the first objective is simple, which is x 1 only and f 2; f 2 objective is comprised of g and h function. So, g function is given as the summation of x i's from i equal to 2 to n and this is multiplied by n divided by n minus 1.

Now, h is also under the square root of f 1 divided by g. This particular problem is solved for 30 number of variables. The characteristic of this problem is that, it has a convex pareto optimal front.

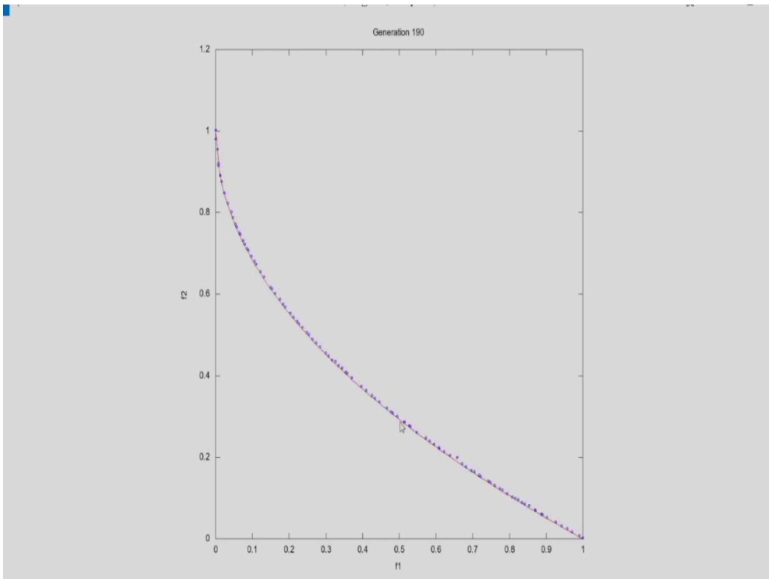
(Refer Slide Time: 41:57)



So, let us see the result of NSGA II on ZDT 1 problem. Here we can see that this is the initial population and the solutions are randomly generated. And the and these solutions are now shown in the objective space of f 1 and f 2. In these simulations, we will be showing the solution in the objective space only.

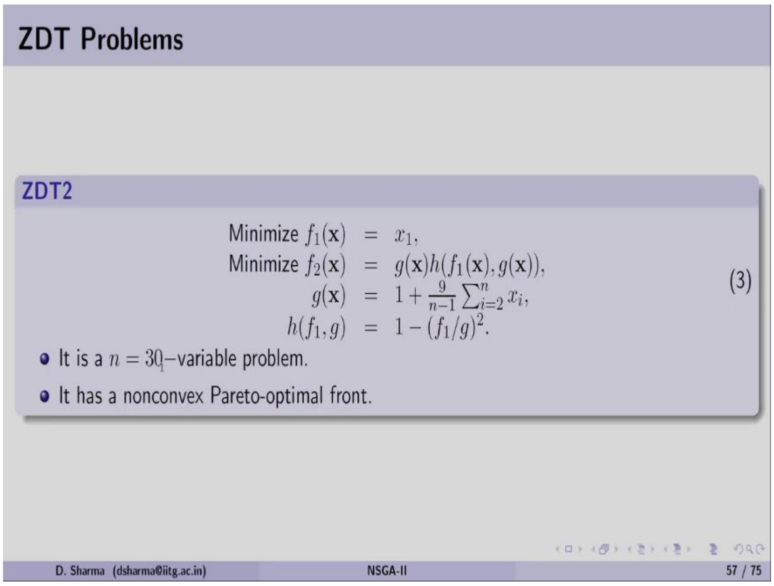
The red line in this particular figure represents the pareto optimal front for the given problem. Since it is a mathematical problem, so we already know where is the pareto optimal front. So, after 200 generation, we can see that the solutions are well converged to the pareto front and they are also distributed from one corner to the another corner. So, let us see the simulation of NSGA II on ZDT 1 problem.

(Refer Slide Time: 43:01)



So, as we can see that the solutions are distributed, they are slowly moving towards the pareto optimal front. And when we are close to the pareto optimal front, they converges little slowly as compared to their initial phase. And finally, all these solutions will be converge to the pareto optimal front as we can see after 200 generation.

(Refer Slide Time: 43:33)



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

$$Minimize \quad f_2\,(x) \;=\; g(x)h(f_1(x), g(x))$$

$$g(x) \;=\; 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$$

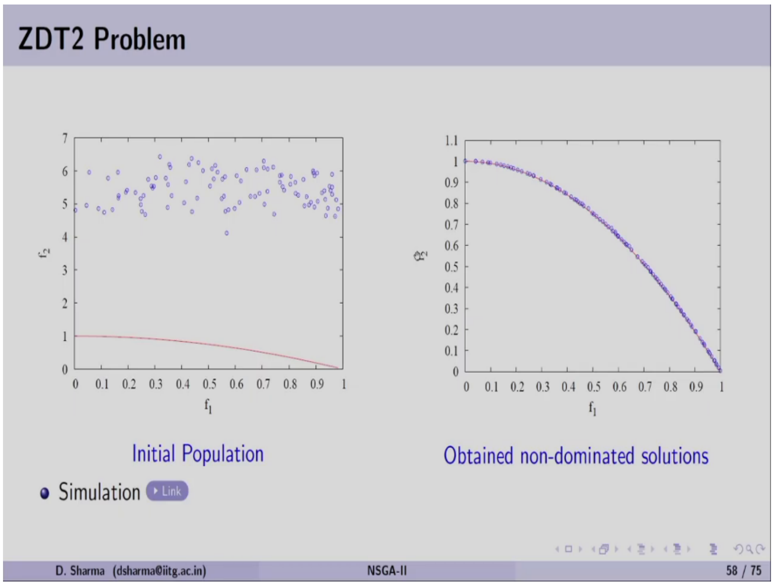
$$h(f_1, g) \;=\; 1 \;-\; \left(\frac{f_1}{g}\right)^2$$

So, this is the simple problem which is very well solved by z by the NSGA II. Now, we will take the another problem that is called ZDT 2. The f 1 objective remains the same, the format of f 2 remains the same. So, the g is also same as ZDT 1; but h has been changed.

So, we can see that 1 minus; now we have f 1 divided by g square and in this particular problem, this h will make our pareto optimal front non-convex. So, as we remember, there are many classical optimization techniques or methods which cannot be used for non-convex pareto optimal front.

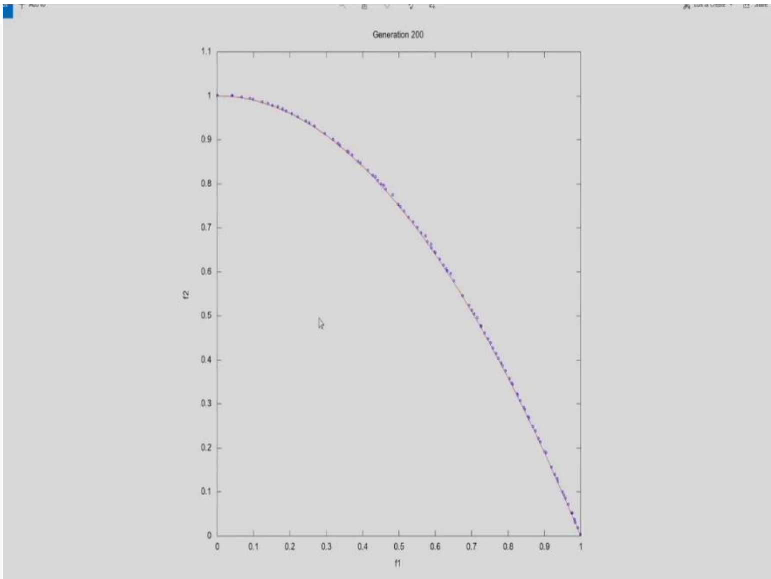
So, now we have to see whether NSGA II can solve this problem or a not. We have n equals to 30 variable for the given problem.

(Refer Slide Time: 44:26)



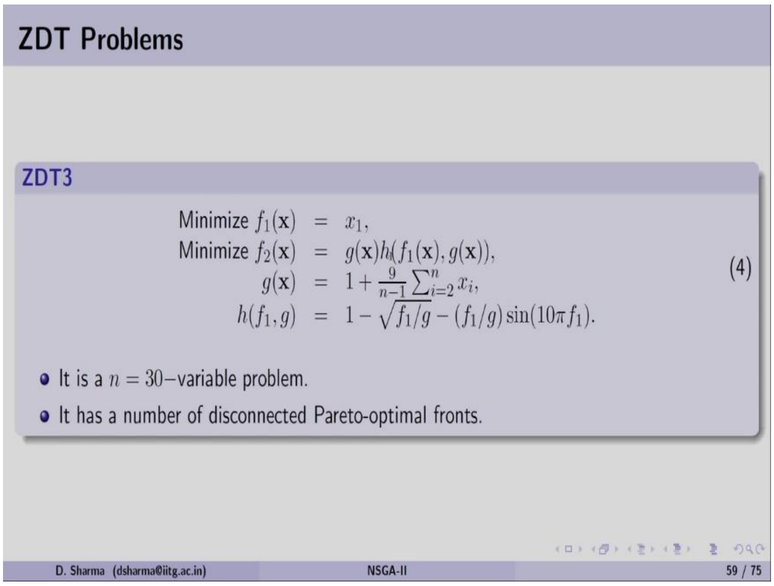
Again we can see the distribution of the solution, which are generated randomly under the initial population. And after 200 generations, we can see the solutions are distributed or converged to the pareto optimal front and also distributed are on the pareto optimal front. Let us see the simulation. How these solutions are converged to the pareto front?

(Refer Slide Time: 44:58)



As we can see the group of solutions they are converging slowly and the algorithm phase no problem in converging for the non-convex pareto optimal front. And we can see at the end of 200 generation, all the solutions are converged to the pareto optimal front. So, we get a we, the solutions are converged as well as distributed on the pareto front.

(Refer Slide Time: 45:29)



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

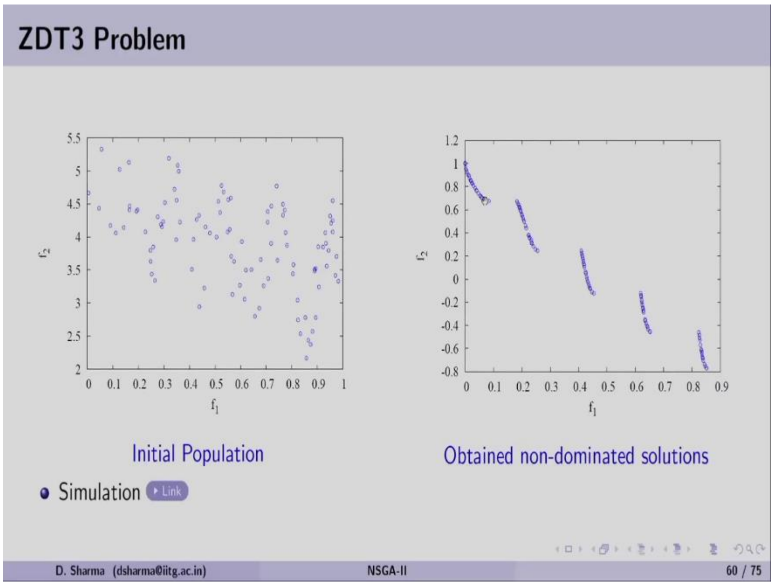
$$Minimize \quad f_2(x) = g(x)h(f_1(x), g(x))$$

$$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$$

$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} - \left(\frac{f_1}{g}\right) sin\,(10\pi f_1)\,.$$

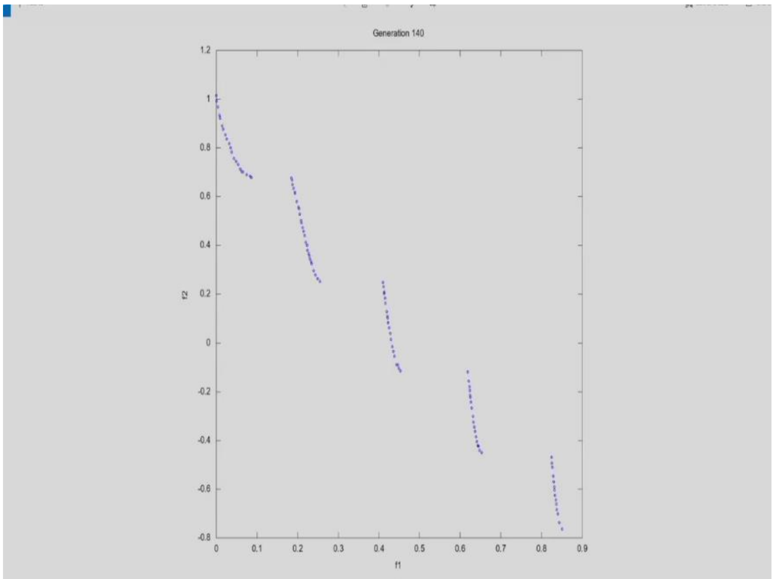
The next problem is the ZDT 3 problem; f 1 and f 2 formats are the same, similarly g is also the same. But in case of h function, we have under root of f 1 g factor as well as we have the sin. This problem is also solved for 30 number of a variable; the characteristic of this problem is, it has number of disconnected pareto optimal front.

(Refer Slide Time: 46:01)



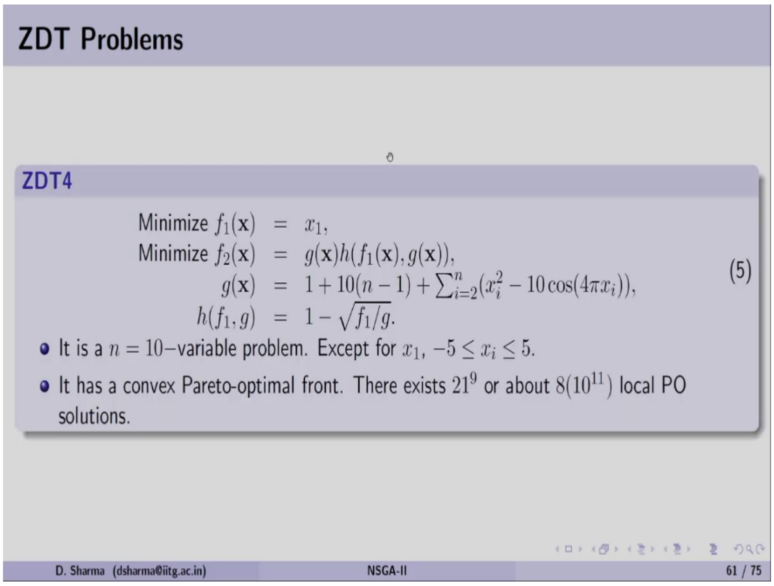
So, this is the initial population which is generated; as we can see in front in f 2 which is y axis, the solutions are distributed in the range of 2 to 5.5. After 200 generation, these solutions are converged to the pareto optimal front. And as we can see, we have these this particular front which is disconnected. Let us see the simulation of the algorithm on ZDT 3 problem.

(Refer Slide Time: 46:32)



So, from the beginning itself, the solutions are distributed on the different front and after few generations, these solutions are converged to the pareto front as well as they have distributed in their respective regions. So, we will see one more time, these solutions have already converged and now distributed. So, as we can as we understand that, NSGA II even can solve the problem which has disconnected pareto optimal front.

(Refer Slide Time: 47:07)



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

$$Minimize \quad f_2(x) = g(x)h(f_1(x), g(x))$$

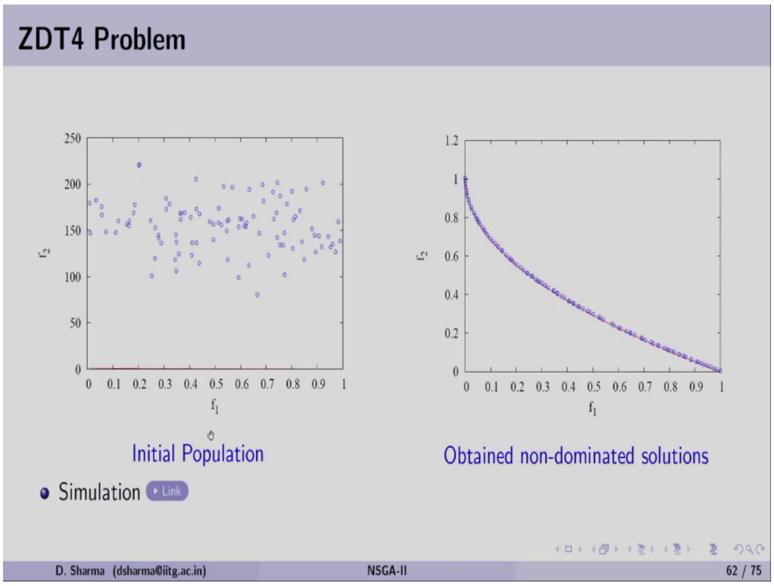
$$g(x) = 1 + 10(n-1) + \sum_{i=2}^{n} (x_i^2 - 10\,cos(4\pi\, x_i))$$

$$h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}}.$$

Now, we are moving to the fourth problem, which is called ZDT 4 problem. Here the format of f 1 and f 2 remain the same; but the g function has been changed. As we can see that x i square minus 10 cos of 4 pi x i this factor has been added; h is again 1 minus under root of f 1 by g. This problem we have considered for 10 variable, except for x 1 all variables will be lying from minus 5 to 5; meaning x 1 will be lying from 0 to 1 and rest of the solution will be lying from minus 5 to plus 5.

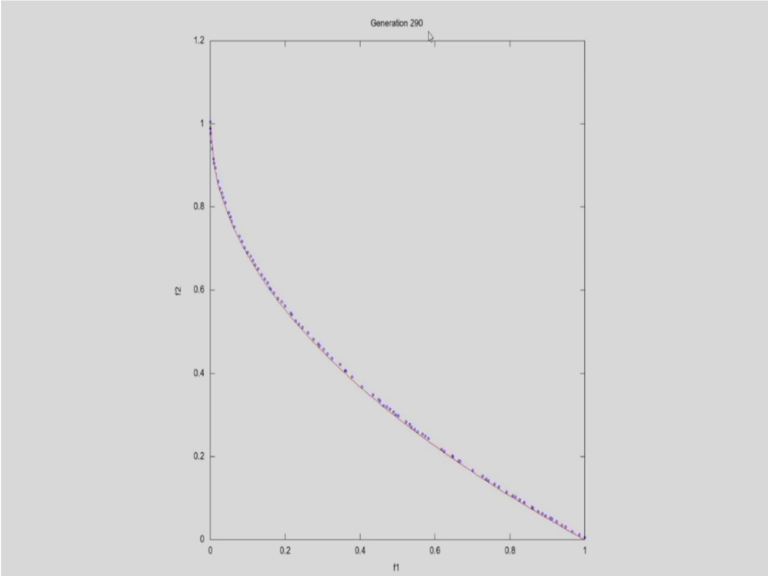
It has a convex Pareto optimal front and there exist 21 to the power 9 or about 8 into 10 to the power 11 local Pareto optimum solution. So, from this particular characteristic; we can understand that this problem is difficult to solve, since it has many local Pareto optimal solutions. So, as we can see in the initial population, the solutions are starting very far from the Pareto optimal front.

(Refer Slide Time: 48:21)



And after 200 generation, we can see the solutions are well converged to the front. We will see the simulation now.

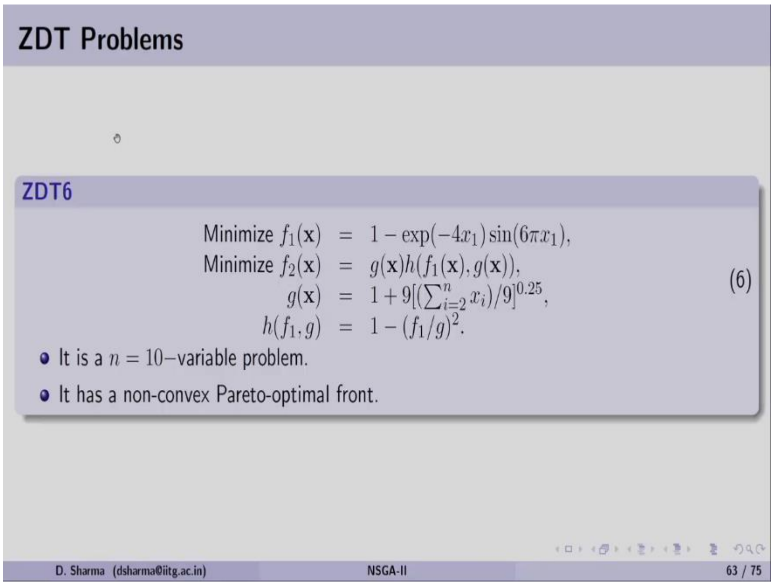
(Refer Slide Time: 48:44)



As can be seen that many of the solutions are on the corner on the one of the corner; once the solution are coming closer to the Pareto front, these solutions get distributed along the front. This for this particular problem, we are running NSGA II for 300 generation; as we can see that the solutions are converged and on the Pareto front.

Now, here again we see that the solutions which were approaching towards the Pareto front from the corner, now they are distributed along the Pareto optimal front.

(Refer Slide Time: 49:25)



$$Minimize \quad f_1(x) = 1 - exp(-4x_1)sin(6\pi x_1)$$

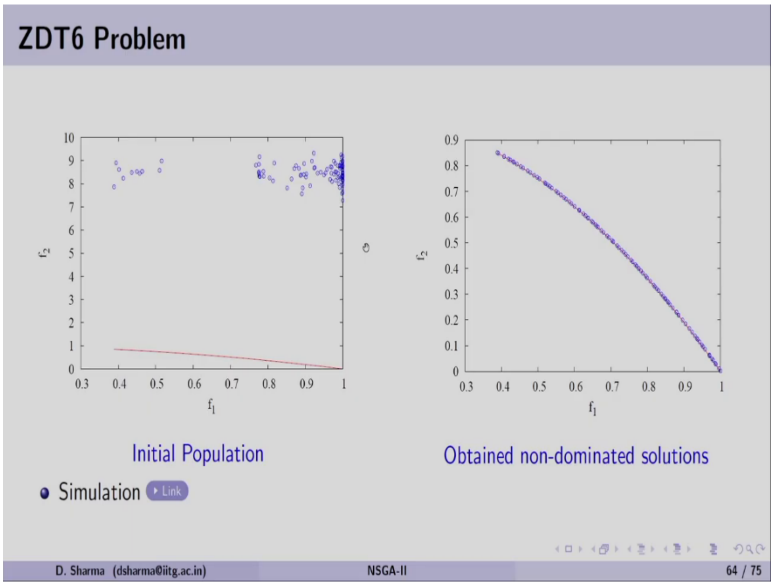
$$Minimize \qquad f_2\,(x) \;=\; g(x)h(f_1(x), g(x))$$

$$g(x) \;=\; 1 + 9\left[\frac{(\sum_{i=2}^{n} x_i)}{9}\right]^{0.25}$$

$$h(f_1, g) \;=\; 1 - \left(\frac{f_1}{g}\right)^2.$$

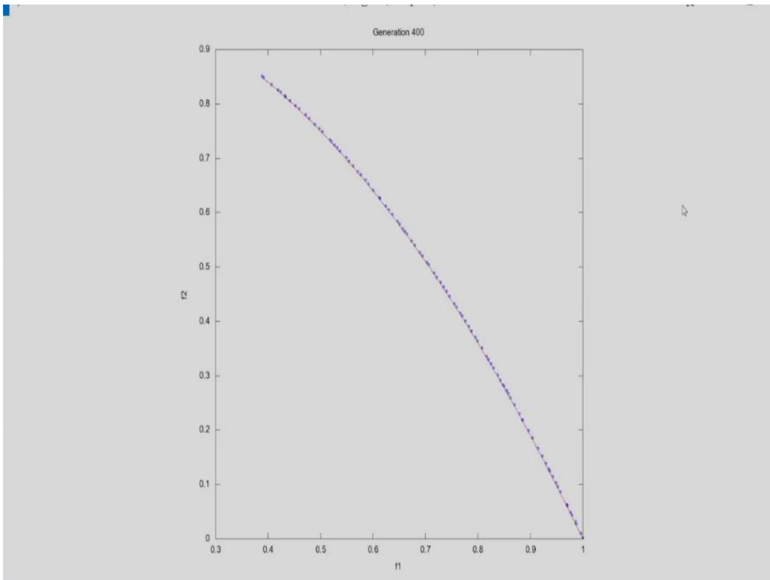
Under this class of ZDT problem, we have the last problem which is ZDT 6. For this particular problem, the f 1 objective is changed here, which incorporate exponential term as well as sin term, and f 2 objective has the same format. And the function g has been changed, and h is 1 minus f 1 g square. For the given problem, 10 number of variables are considered and this particular problem has a non-convex Pareto optimal front.

(Refer Slide Time: 50:05)



So, this is the initial population generated for the given problem and we can see that many of the solutions are generated on the one side, which we can say that the skewed population. And after running this particular algorithm for many generations, we can see that the solutions are very well distributed on the Pareto optimal front. Let us see the simulation now.

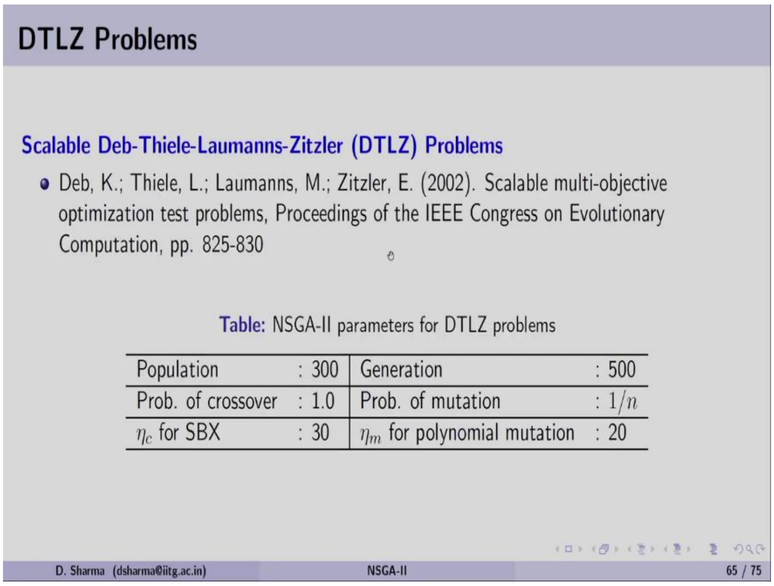
(Refer Slide Time: 50:34)



Here we can see the solutions are moving along the corners as well as some solution in the middle. As and when we are close to the Pareto optimal front, these solutions are distributed and slowly and slowly these solutions are moving towards the Pareto front. We have run this particular algorithm for more number of a generation as we can see on top; it is already 400 generations.

The problem which we have solved using NSGA II, all of them are by objective problem. Just to show the significance of the same algorithm or we can say the performance of the given algorithm, we are going to solve some three objective problems.

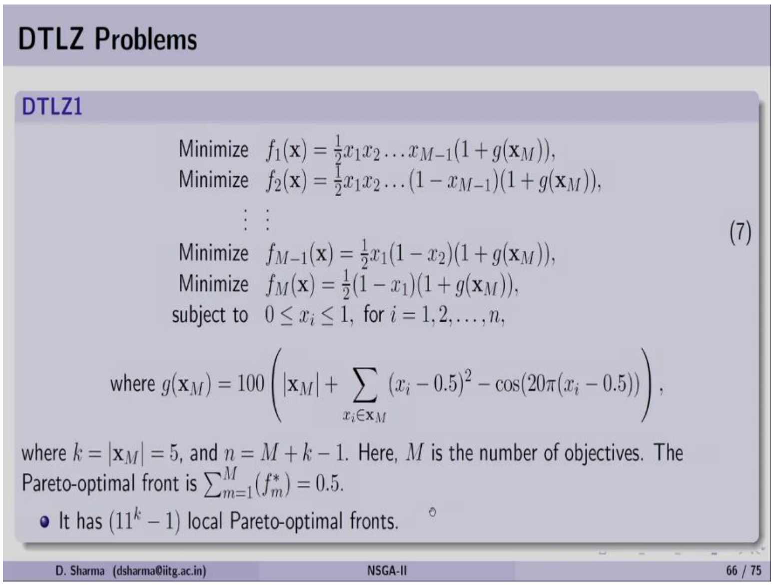
(Refer Slide Time: 51:30)



So, the another class of problem which we have chosen for the performance assessment is called DTLZ problem. Now, we can see here these DTLZ problem, the name came from the surname of the authors of these problems and the details of the paper are given here. These problems are called scalable problem; it is only because we can change the number of objectives.

We can consider two, three and many more number of objectives to test our algorithm. The NSGA II parameters for DTLZ 2 problem, we can see population is kept 300, generation is 500, probability of crossover is 1, probability of mutation is 1 by n, SBX eta c value is 30, and eta m for polynomial mutation is 20.

(Refer Slide Time: 52:22)



$$Minimize \qquad f_1(x) = \frac{1}{2}x_1x_2 \ldots x_{M-1}(1 + g(x_M))$$

$$Minimize \qquad f_2(x) = x_1x_2 \ldots (1 - x_{M-1})(1 + g(x_M))$$

$$\vdots \, \vdots$$

$$Minimize \qquad f_{M-1}(x) = \frac{1}{2}(1 - x_2)(1 + g(x_M))$$

$$Minimize \qquad f_M(x) = \frac{1}{2}(1 - x_1)(1 + g(x_M))$$

$$subject\ to \qquad 0 \le x_i \le 1, \qquad for\ i = 1,2, \ldots, n,$$

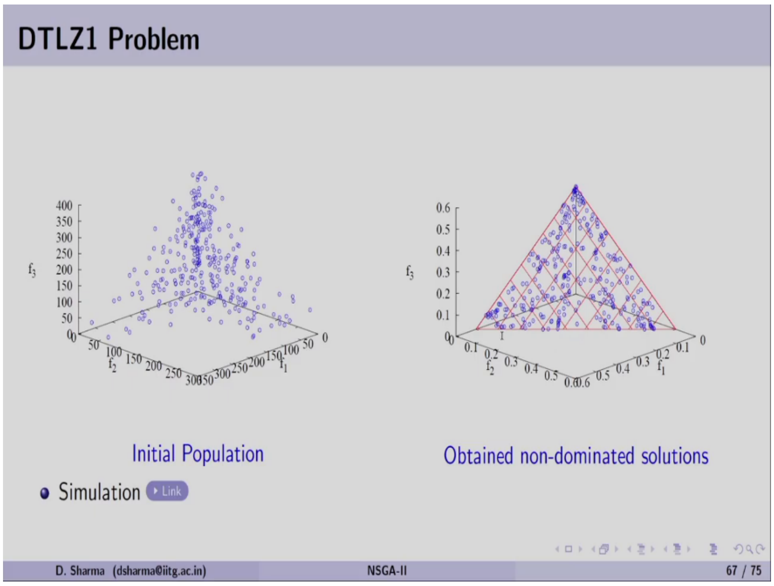
$$where \qquad g(x_M) = 100\left(|x_M| + \sum_{x_i \in x_M} (x_i - 0.5)^2 - cos\big(20\pi\,(x_i - 0.5)\big)\right),$$

Let us discuss the DTLZ 1 problem. As we can see, it is a scalable problem. So, we can write say M number of objective. So, it is starting from minimizing f 1, f 2 till f M and M can be any number. So, the format of this f 1, f 2, f M we can see that we have x 1 x 2 till X M and then we have a one function based on g and there is a parameter called X M.

In this particular problem, the all variables are lying between 0 to 1 and the function g is given a here. As we can see it depends on X M value and inside the bracket we have x i 0 minus 0.5 square as well as we have a cos term. For ZDT 1 problem, it was suggested to take k equals to the cardinality of X M which is 5, and the number of variable for the given problem is M plus k minus 1.

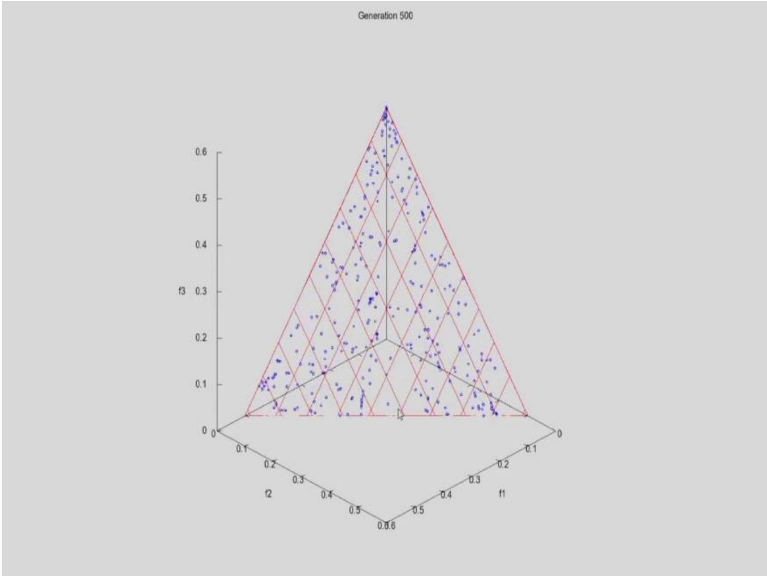
M is the number of objective, k is 5 for DTLZ 1 problem and minus 1. The Pareto front of this particular problem is the summation of all objective equals to 0.5. So, it is going to give us a plane now. The characteristic of this problem is, we are going to have 11 to the power k minus 1 local Pareto optimum front. So, as from the problem it is evident that, this problem is difficult to solve.

(Refer Slide Time: 54:07)



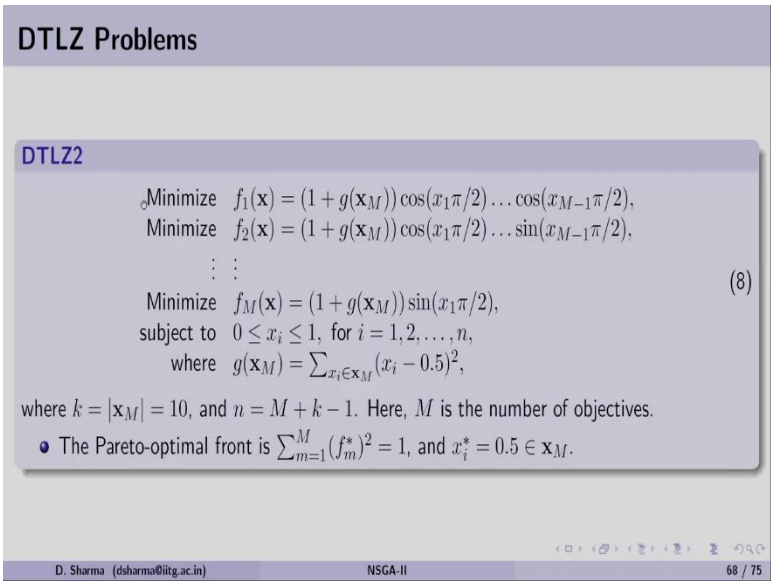
So, this is the initial population where NSGA II started and it is quite far; because we remember that the Pareto optimal front will be lying at 0.5. And after running for 500 generation, we can see that the solutions are converged to the Pareto front and they are also distributed along that Pareto optimal surface. So, now since it is three objective, we call it as a Pareto optimal surface.

(Refer Slide Time: 54:45)



If we see the simulation of NSGA II for DTLZ 1 problem; we can see that the solutions for are coming from very far, and now they are started converging on the Pareto surface which is a plane in this given problem And once these solutions are converged here; we can see that, the we are that crossover and mutation are generating new solutions on the Pareto surface and based on the crowding distance, the solutions were selected.

(Refer Slide Time: 55:20)



$$Minimize \qquad f_1(x) = (1+g(x_M))cos\,(x_1\pi/2) \;\ldots\; cos\,(x_{M-1}\,\pi/2),$$

$$Minimize \qquad f_2(x) = (1+g(x_M)\;)cos(x_1\pi/2)\ldots sin(x_{M-1}\,\pi/2)$$

$$\vdots \quad \vdots$$

$$\begin{aligned} Minimize \qquad & f_M(x) &= (1 + g(x_M))sin(x_1\pi\,/2)\,) \\ subject\ to \qquad & 0 \leq\ x_i \leq\ 1, \qquad & for\ i = 1,2, \dots ,n, \\ where, \qquad & g(x_M) &= \sum_{(x_i \in x_M)} (x_i - 0.5)\hat{}2 \end{aligned}$$

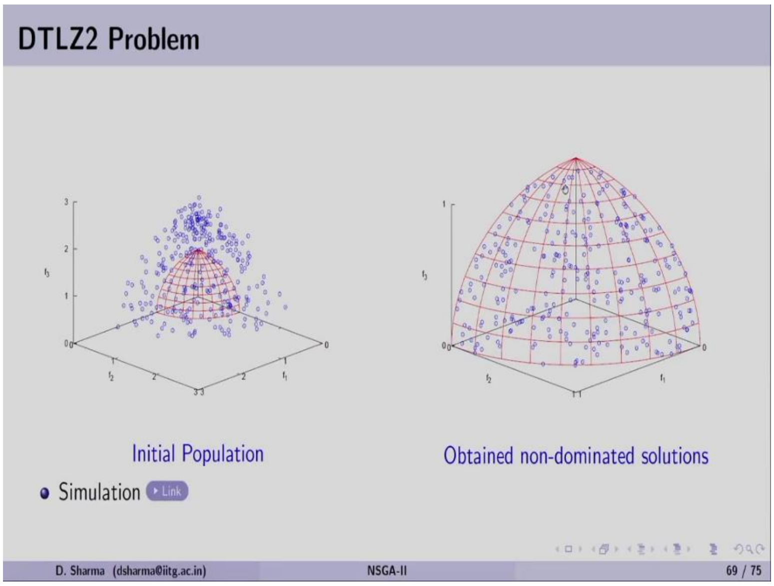
$$where\ k = |x_M| = 10, \qquad and \quad n = M + k - 1$$

$$\sum_{m=1}^{M} (f_m^*)^2 \ = 1, \qquad and \qquad x_i^* \ = \ 0.5 \ \in \ x_M$$

Let us move to the second problem which is DTLZ 2 problem; since it is a scalable, the format of f 1, f 2 and f M are given. In this particular problem, cos and sin terms are included. Again the variable bounds for all the variable is are given as 0 from 0 to 1 and the function g is given as x i minus 0.5 square.

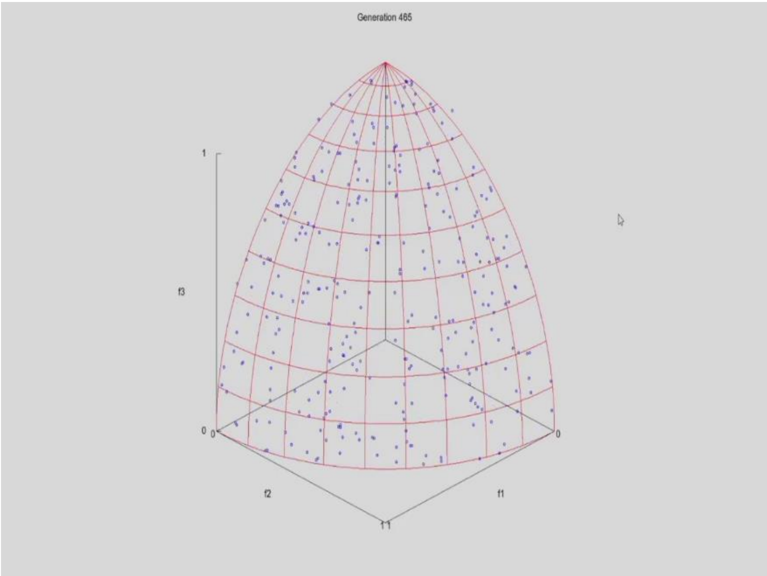
For DTLZ 2 problem, k is considered as 10, number of variable is equals to M plus k minus 1 and where M is the number of objective and k is taken 10 for DTLZ 2 problem. The Pareto optimal front is f star m square, meaning it is going to be hyper sphere for us. And all the Pareto optimal solution will be at 0.5, x i equals to 0.5.

(Refer Slide Time: 56:24)



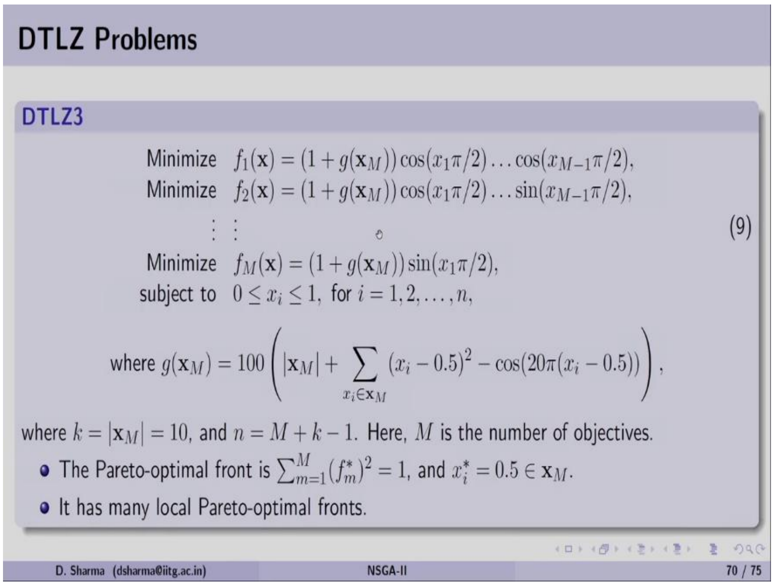
Now, let us solve this problem, initially the solutions are generated randomly as we can see on the figure on the left hand side. And after many after the 500 generation, we can see on the right hand side; the solutions are converged to the Pareto optimal front. We will see the simulation for the given problem now.

(Refer Slide Time: 56:48)



Now, we can see the problem is relatively simple and therefore, the solutions quickly converged to the Pareto front. Once these solutions are converged here, we can see the solutions are keep on changing their position; it is only because all the solutions are rank 1 and we are selecting the solution using crowding distance now. So, for this particular problem, the algorithm converges quickly.

(Refer Slide Time: 57:17)



$$Minimize \qquad f_1(x) = (1 + g(x_M))cos\,(x_1\pi\,/2) \;\; \ldots \;\; cos\,(x_{M-1}\,\pi\,/2),$$

$$Minimize \qquad f_2(x) = (1 + g(x_M)\;)cos(x_1\pi\,/2) \ldots sin(x_{M-1}\,\pi\,/2)$$

$$\vdots \;\; \vdots$$

$$Minimize \qquad f_M(x) \;\; = (1 + g(x_M))sin(x_1\pi\,/2)\;)$$

$$subject\ to \qquad 0 \le \; x_i \le \; 1, \qquad for\ i = 1,2, \ldots\ , n,$$

$$where, \quad g(x_M) \; = \; 100\;(|x_M| \; + \sum_{x_i \in x_M} (x_i - 0.5)\text{^}2 \;\; - \; \cos(20\pi\;(x_i \; - \; 0.5))) \quad ,$$

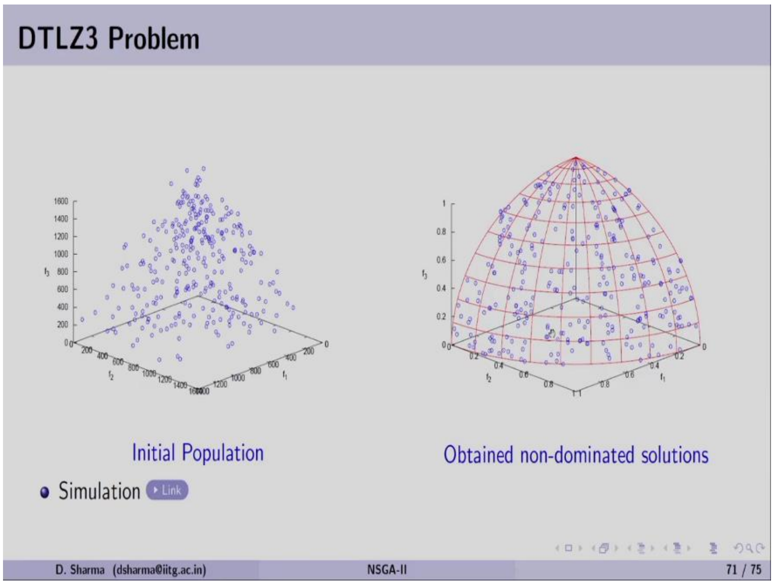
$$where \quad k = |x_M| = 10, \qquad and \qquad n = M + k - 1$$

$$\sum_{m=1}^{M} (f_m^*)^2 = 1, \qquad and \qquad x_i^* \; = \; 0.5 \; \in \; x_M$$

Now, coming to the third problem, which is the DTLZ 3 problem. The format of f 1, f 2, f m objectives are given in equation number 9; variable bound bounds are the same as 0 and a 1 and the function g M is taken similar to the DTLZ 1. Here k is considered as 10, number of variable will become M plus k minus 1; the Pareto front will be the same as DTLZ 2.

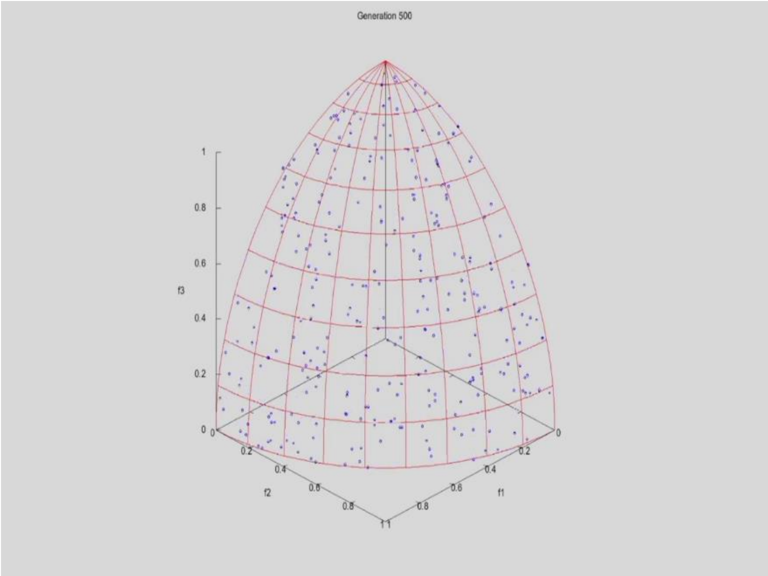
And in this particular DTLZ 3 problem, we will have many local Pareto optimal front. So, meaning that, we are adding a complexity into DTLZ 2 problem by introducing many local Pareto optimal front.

(Refer Slide Time: 58:06)



Now, the initial population is generated and in the objective space, we can see that the solutions are generated and they are very far from the Pareto surface. After running the algorithm for 500 generation, we can see the solutions are distributed; first converged to the Pareto surface, and they are also distributed along the surface. Let us see the simulation for the given problem now.

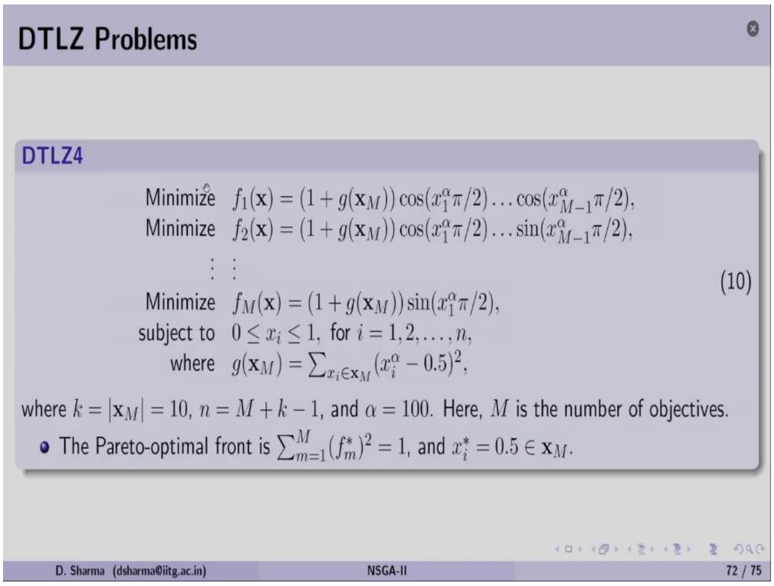
(Refer Slide Time: 58:38)



In this particular problem, we can see the solutions are moving from very far, and slowly and slowly they are converging towards the Pareto front. So, after 200 generation, these solutions converged to the Pareto optimal front. So, as we can see the solutions are already converged.

So, in this case what we observe that, initially since there were many local Pareto optimal front; so that is why the solution are slowly moving to, slowly moving towards the Pareto surface. And once they are closed, they are now distributing over the Pareto optimal surface.

(Refer Slide Time: 59:24)



$$Minimize \quad f_1(x) = \big(1 + g(x_M)\big)\, cos(x_1^\alpha \pi/2) \ldots cos(x_{M-1}^\alpha\, \pi/2)$$

$$Minimize \quad f_2(x) = \big(1 + g(x_M)\big)\, cos(x_1^\alpha \pi/2) \ldots sin(x_{M-1}^\alpha\, \pi/2)$$

$$Minimize \quad f_M(x) = \big(1 + g(x_M)\big)\, sin(x_1^\alpha \pi/2)$$

$$subject\ to \quad 0 \le x_i \le 1, \quad for \quad i = 1,2, \ldots, n$$

$$where \quad g(x_M) = \sum_{(x_i \in x_M)} (x_i^\alpha - 0.5)^\wedge 2,$$

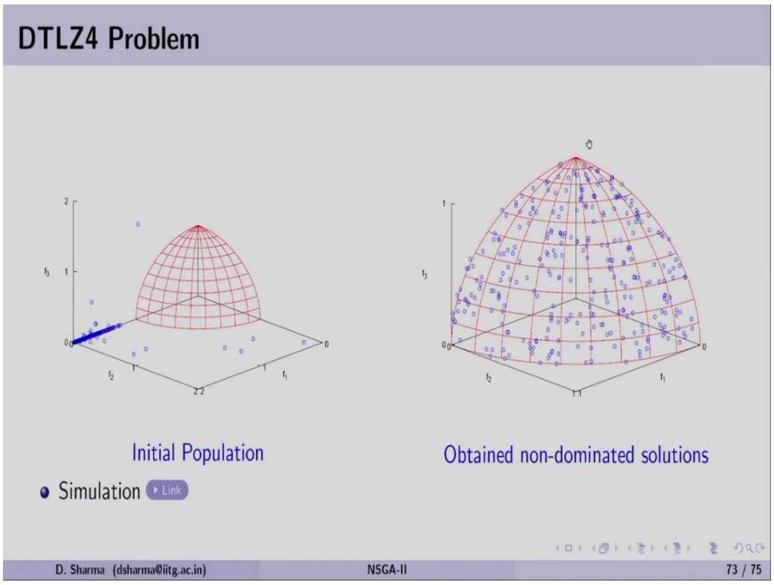
$$where\ \ k = |x_M| = 10, n = M + k - 1, \quad and\ \alpha = 100$$

$$\sum_{m=1}^{M} (f_m^*)^2 = 1, and\ x_i^* = 0.5 \in x_M$$

We have come to the last problem in the DTLZ family; although there are many DTLZ problem, we will show the simulation of NSGA II for this DTLZ 4 as a last problem. Now, we can see that this is the same as DTLZ 1, DTLZ 2 problem; however, this alpha term is included in every equation of f 1, f 2 and f M, because of that there is a that incorporate some complexity for the algorithm to converge.

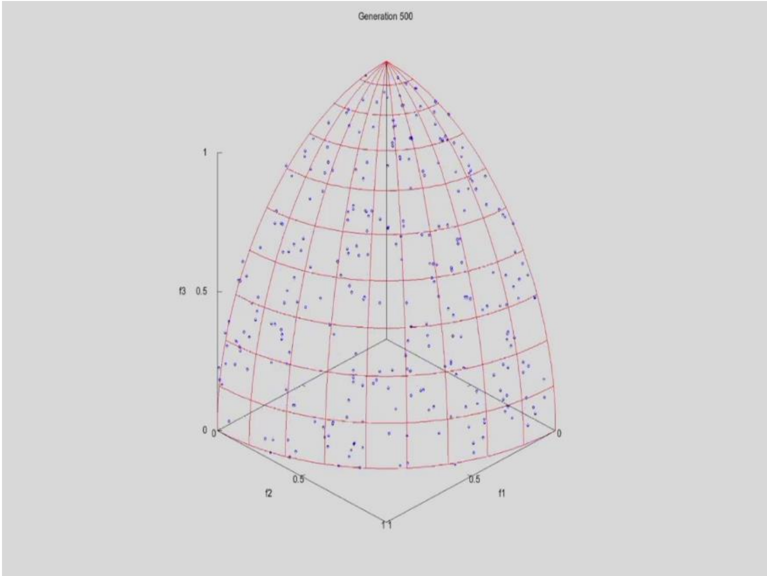
Variable bound is again lying between 0 to 1 and g M function is given as x i to the power alpha minus 0.5 square. For the given example, given problem we are taken k equals to 10, number of variable as M plus k minus 1 and alpha is taken as a large value. Now, the Pareto optimal front is the same as DTLZ 2, which is a hyper sphere and let us see how we are going to solve this problem.

(Refer Slide Time: 60:27)



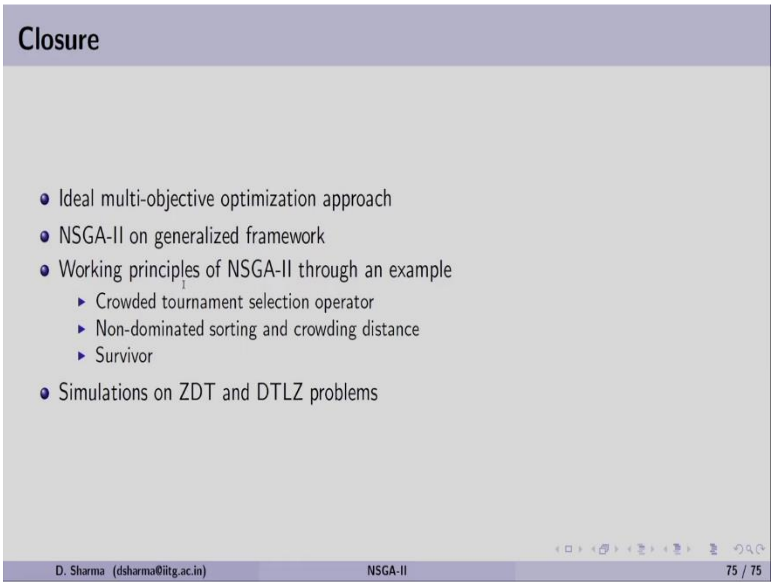
So, initially these solutions are generated and as we can see, many of the solutions are on one of the corner. And finally, after 500 generation, the solutions are converged to the Pareto front and they are distributed as well.

(Refer Slide Time: 60:49)



So, let us see the simulation of NSGA II for the given problem. So, as we can see, although we started from the corner; these solution quickly converged to the Pareto front. And once they converged, now they are distributing the solution along the Pareto surface. Now, till 500 generation, these solutions are changing their position; because all of them are rank 1 solution.

(Refer Slide Time: 61:12)



With this simulation, we have come to the closer of the session on NSGA II. In this particular session, we discussed about the ideal multi objective optimization approach;

because in this approach, when we are going to solve the multi objective problem, we will be generating the multiple Pareto optimal solutions on the surface.

After generating those solutions, using the higher level of information; we can always choose the desired solution. But the question comes, how we can generate those multiple solutions on the Pareto optimal front? And therefore, EC techniques shows upper hand over the other algorithm; it is only because we work with the number of solution, which is called as a population with EC techniques.

By appropriately changing the fitness in NSGA II, it was done using the non dominated sorting by assigning the rank and sorting the solutions in fronts and thereafter, calculating the crowding distance. Once we change the fitness, there are few more changes that has been done with NSGA II; that are for example, the crowded binary tournament selection and the survival stage, how we are going to select best n solution.

So, in this particular session, we understood this NSGA II through an example. So, the working principle, all the operators which we discussed here that includes the crowded tournament selection, non-dominated sorting and crowding distance, and the survivor. The performance of NSGA II is tested on two objective and three objective mathematical problems; these problems are bench marked, these problems are considered as benchmark problems.

Whenever we come up with the new algorithm, we always test these algorithm on these benchmark problems, so that we can assess the performance. And once they are good, we can even extend these algorithm to solve some real world problem. With this hand calculation, simulation, and graphical understanding of NSGA II, I conclude this session.

Thank you very much.