Optimization Methods for Civil Engineering

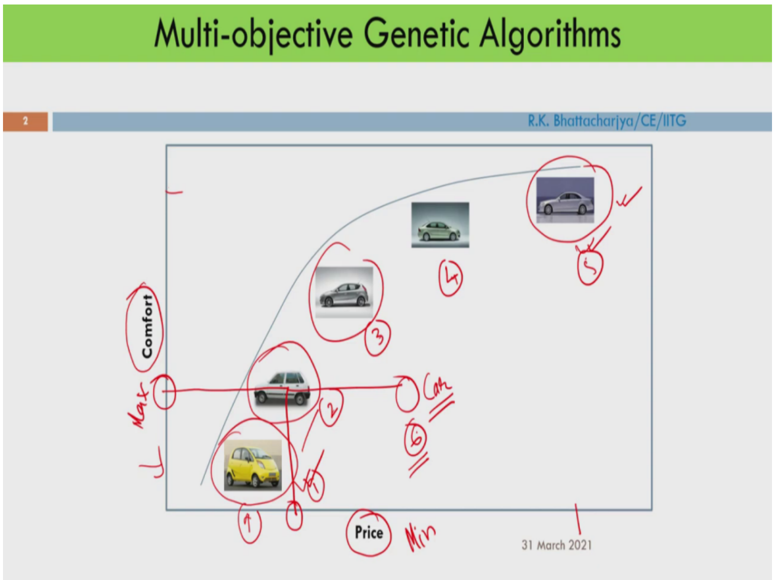
Dr. Rajib Kumar Bhattacharjya
Department of Civil Engineering
Indian Institute of Technology, Guwahati

Lecture - 29
Multi-Objective Genetic Algorithms

Hello student, welcome back to the course on Optimization Methods for Civil Engineering. So, today we will discuss Multi Objective Genetic Algorithms; already we have solved optimization problem using genetic algorithm, so we have mainly solved the single objective genetic algorithm.

So, today we will solve multi objective genetic algorithm. Now, let us discuss one multi objective problem; suppose I have shown here a problem, suppose I would like to purchase a car, ok. .

(Refer Slide Time: 01:07)



So, in that case, so what will happen? I have different options or I can say these are some of the solution; suppose this is a Nano car and this is suppose Maruti Alto and then I 20 something like that and we have a very luxury car, Honda city or something like that.

Now, if I see at this car, suppose if I say the price point of view. So, my best solution is the this one. So, I can purchase this car at a very less cost basically less price. So, I can purchase this car; but if I look from comfort point of view, this is not a very good solution or this is not a very good idea to purchase this car.

Similarly, if I purchase this car, may be your from comfort point of view or luxury point of view, this cars will be the best option or it is a better option; but if you look at from the cost point of view, this is not a very good solution. So, this is the inferior solution in terms of cost.

So, therefore, what is my objective here? So, my objective here is that, I would like to minimize; I would like to minimize price or cost or I would like to maximize your comfort. So, if I look at my first objective, then this is the best solution among this available solution and if I look at from the cost point of view, so this solution will be the better solution or best solution.

Now, let us compare two solutions here; suppose if I compare these two solution, suppose if I say this is solution 1 and this is solution 2, this is solution 3, this is 4 and this is solution 5. Now, if I compare solution 1 and 2; can I say that solution 1 is better than solution 2 or can I say that solution 2 is better than solution 1?

So, if I compare these two solution, I cannot actually tell that whether solution 1 is better or whether solution 2 is better; because in terms of cost solution 1 is better, but in terms of comfort solution 1 is inferior than solution 2. Similarly, in terms of comfort, I can say solution 2 is better than solution 1; but in terms of cost solution 2 is inferior than solution 1.

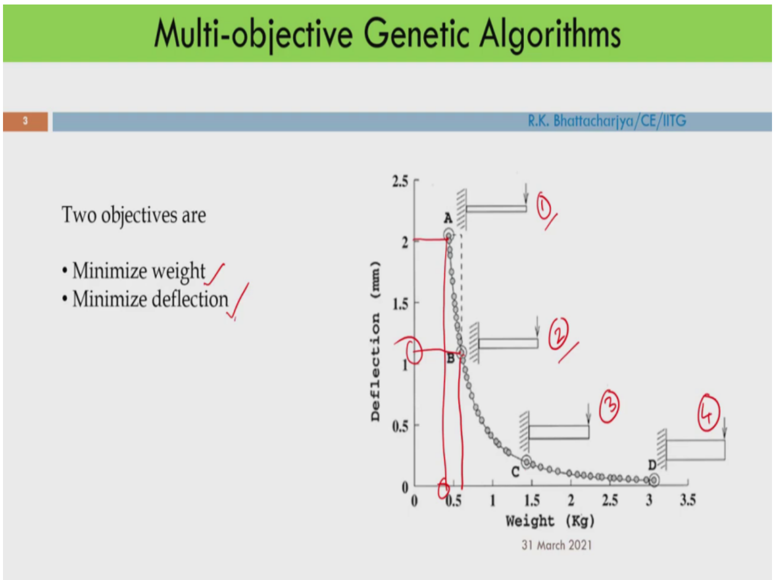
So, therefore, I cannot compare the solutions, ok. So, we call it non dominated solution; that means one solution is not dominating the other solution. Now, if I have a car somewhere here. Suppose this is the; this is this particular solution. So, this is the comfort level of this particular solution and this is the cost level at this particular solution. Now, I have a solution somewhere here; this is another car basically, so another car so.

Now, if I compare this solution maybe this is the 6th solution; now if I compare the solution 6 and solution 2, in that case I can say the solution 2 is better than solution 6. So, in that case I can say that solution 6 is inferior than solution 2. Why? Because in terms of objective 2; that means comfort, solution 2 and solution 6 are equal, ok. But in terms of your price or in terms of cost, solution 2 is much better than solution 6. So, therefore, solution 6 is dominated by solution 2. So, I have a better solution in this case.

But if I compare the other solution, suppose if I compare solution 2 and solution 3, solution 3 or solution 4, or solution 4 or solution 5, or solution 5 or solution 1, I cannot tell actually which one is better solution and which one is inferior solution; because in one objective one solution is better, in another objective the other solution is better.

So, therefore, this solution can be your term as a non-dominated solution; that means these solutions are non-dominated solution, ok. I hope this is clear to you. Now, let us take another example problem. So, in that case, I would like to design a cantilever beam.

(Refer Slide Time: 05:26)



So, what is the objective here? So, my objective is the minimize weight; that means I would like to have minimum weight. Why minimum weight? Because once weight is minimum; that means I will use less material and therefore, cost of the beam will be less. So, you can say that one objective is minimize weight and another objective is minimize deflection.

Suppose what will happen, if you are reducing your section, the depth of the section; then deflection will be more, so in that case sometime that is not acceptable. So, therefore, you can say that, these are conflicting in nature, these two objectives are conflicting in nature.

So, let us compare. So, here also I would like to mark like this, this is solution 1 and this is solution 2, this is solution 3, and this is solution 4. Now, if I compare solution 1 and solution 2; can I tell whether solution 1 is better or solution 2 is better? So, I cannot tell that one because.

So, if I compare solution 1 and solution 2, suppose this is the weight of this solution and this is the deflection level of this particular solution. Similarly, for solution 1 this is the weight and this is the deflection of this particular solution. So, therefore, when I am comparing solution 2 and solution 3; that solution 1 is better in first objective, because the weight is less, ok.

So, weight is less. So, this is solution 1; solution 1 is better than second solution in terms of weight. So, you can say that solution 1 is better in first objective. In case of second objective, solution your 2 is better; because deflection is lesser than solution 1. So, therefore, I cannot compare solution 1 and solution 2 and similarly if you compare solution 2 and solution 3; then also you will not be able to compare these two solution and similarly C and D also.

So, what is happening that if I want to minimize weight, then I have to compromise somewhere else. So, what I will compromise? I will compromise in terms of deflection. So, deflection will be more. And if I want to minimize deflection; that means I would like to have minimum deflection. So, in that case what you are compromising? So, you are compromising in terms of weight.

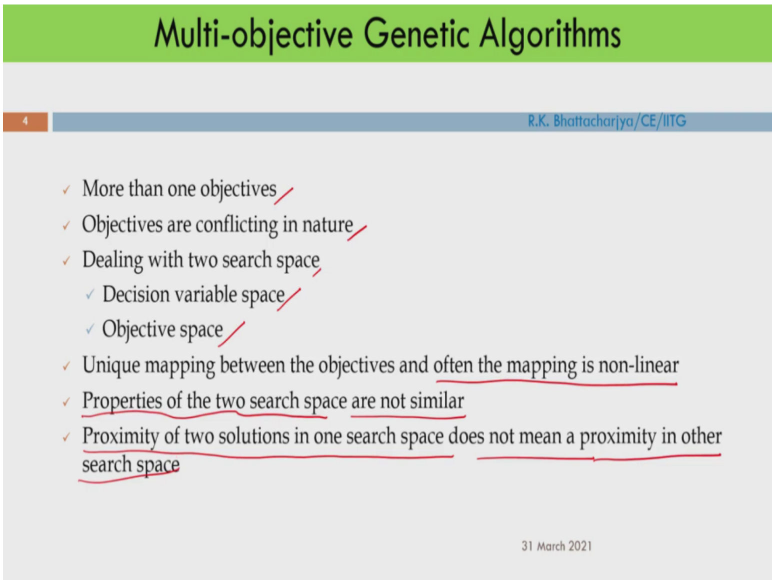
So, you have to give more weights, depth of the section will be more. So, therefore, these two objectives are conflicting in nature; that means if you want to satisfy one objective, you have to compromise on the other objectives, that means there is a trade-off between these two objectives.

Once the objectives are conflicting in nature; so in that case this is a multi-objective optimization problem, that means you will not be able to satisfy both the objectives at the same time, ok. So, if you want to improve in one objective, then you have to sacrifice on the other objectives, ok. So, I hope this is clear. So, this problem is a multi-objective optimization problem.

Now, multi objective, so here I have shown you two objectives, ok. So, in this problem, this is a minimize weight and minimize deflection. On the other example problem when you are purchasing a car, so you want to minimize cost, at the same time you want to maximize comfort.

And these two objectives are conflicting in nature and there are two objective; but there may be some other problem, which will have more than one objective. Now, if your in your problem, if you have more than one objectives. So, in that case, the problem will be termed as multi objective optimization problem.

(Refer Slide Time: 09:19)



So, here I have shown you two objective, but it may be more than two also. So, what is multi objective optimization problem? Multi objective problem will have more than one objective.

So, objectives are conflicting in nature. So, what is conflicting in nature? That means, if you want to satisfy one objective or if you want to make improvement in one objective; in that case you have to sacrifice on the other objectives. Like the if you want to maximize your comfort, then where you have to compromise? You have to compromise in your cost, that means cost of the car will be more.

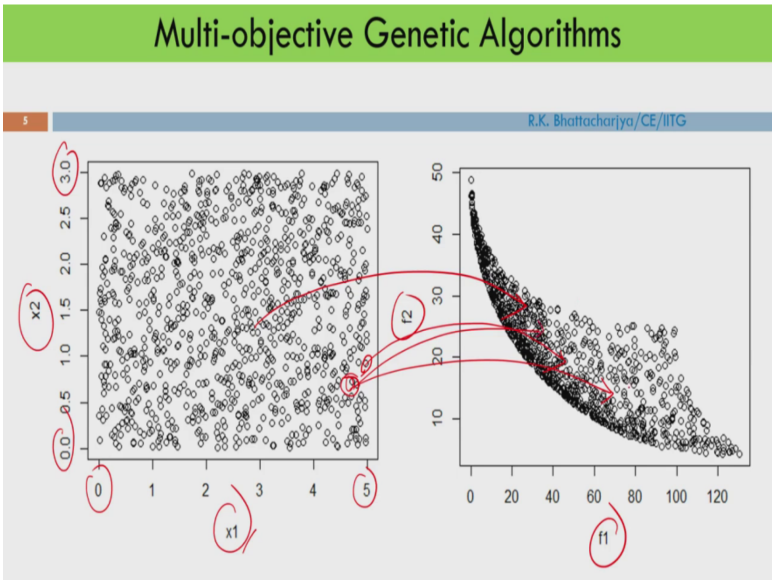
Similarly, when you want to minimize deflection, at that time the depth of the section will be more and you are compromising on the weight of the section, ok. So, therefore, these two objectives are conflicting in nature; that means at the same time you will not be able to satisfy both the objective, that means if you want to minimize cost, then you will not be able to maximize the comfort of that car.

Similarly, if you want to minimize deflection, at the same time you will not be able to minimize the cost. So, therefore, there are two objectives and these two objectives are conflicting in nature. Now, the third one is dealing with two search space. So, dealing with two search space; one is decision variable search space and another one is objective space. So, there are two search space; so one is in the decision variable search space and another is the objective search space.

So, there are unique mapping between the objectives and often the mapping is non-linear. Then properties of two search space; so as I said there are two search space, one is decision search space and one is objective search of and one is objective search space. So, properties of these two search space are not similar; that means if two solutions are closer in one search space, it does not mean that solution will also be closer in the objective search space, ok.

So, proximity of two solutions in one search space does not mean a proximity in other search space, ok. So, that means if two solutions are very close in one search space; it does not mean that the solution will be also means close in the other search space. Similarly, if two solutions are very close in objective search space, it does not mean that solution will also be close in the decision variable search space, ok.

(Refer Slide Time: 11:44)

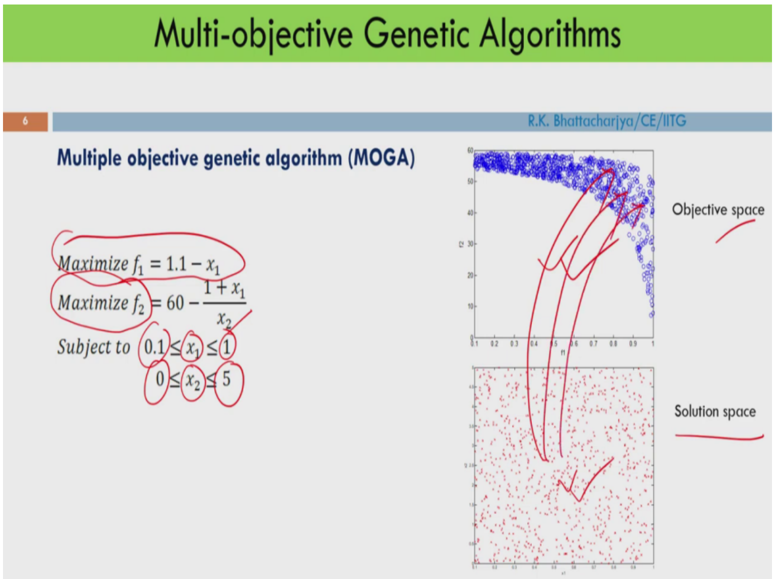


So, here I have shown you two search space. So, one is that variable search space; that means in this problem I have two variables, that is x 1 and x 2. So, x 1 is between 0 and 5 and x 2 is between 0 and 3 and these are some of the solutions what is so randomly generated over this search space. So, these are some of the solutions randomly generated over the search space and you can see. So, these are, there is a mapping between these two search space.

So, this is the objective search space; that means for a particular solution, suppose this is a solution which is x 1 equal to around 5 and x 2 equal to 1 and corresponding there is a point somewhere here and which will basically show you what is the value of f 1 and f 2. So, there is a mapping between these two search space and as I said, so and most of the time this mapping is non-linear mapping, ok.

And any particular two solution, suppose this solution and this solution is very close in the search space. So, these two solution may be at different position or may not be close in the other search space. We are dealing with two search space in case of a multi objective optimization problem.

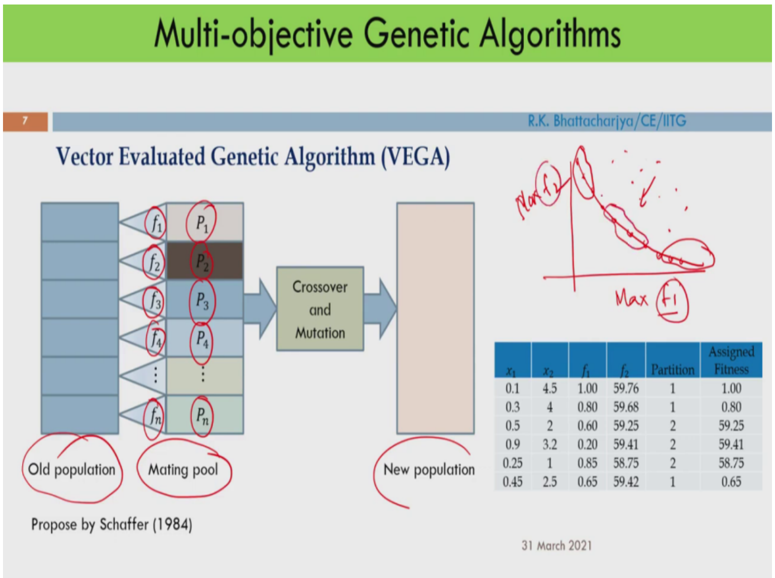
(Refer Slide Time: 13:00)



So, here I have shown an example of multi objective optimization, multi objective optimization problem. So, here there are two objectives that I would like to maximize; maximize f 1 that is 1. 1 minus x 1 and similarly I would also like to maximize f 2, which is equal to 60 minus 1 plus x 1 by x 2.

And x 1 is varying between 0.1 and 1 and x 2 is varying between 0 and 5. So, you can see. So, this is the objective space. So, this is the objective space and this is the solution space and there is always a non-linear mapping or there is a mapping between solution space and objective space.

(Refer Slide Time: 13:45)



Now, let us see how we can solve a multi objective problem using genetic algorithm. So, already we have learned what is genetic algorithm. So, we know what is, what are the steps of genetic algorithm? That means we are initializing solution and then we are calculating the fitness value of the solution, then the solution will go through the genetic operators.

So, what are the genetic operators? The operators are selection operator, crossover operator, and mutation operator and then we are also using elitism. So, this is your simple genetic algorithm. Now, let us modify the simple genetic algorithm and try to solve a multi objective optimization problem.

Now, what is our objective? In this case the objective is, I would like to find out the non-dominated solution. Suppose I have an optimization problem. So, I would like to maximize f 1 and I would like to maximize f 2. So, I have two objective and finally, this should be the pareto optimal front.

So, what is pareto optimal front? So, here all the solutions will be the non-dominated solution, ok. So, these solutions will be non-dominated solution and there is no solution on this other side. So, initial solution will be somewhere here; an idea is that I would like to push all these solution towards pareto optimal front. How we can do that one?

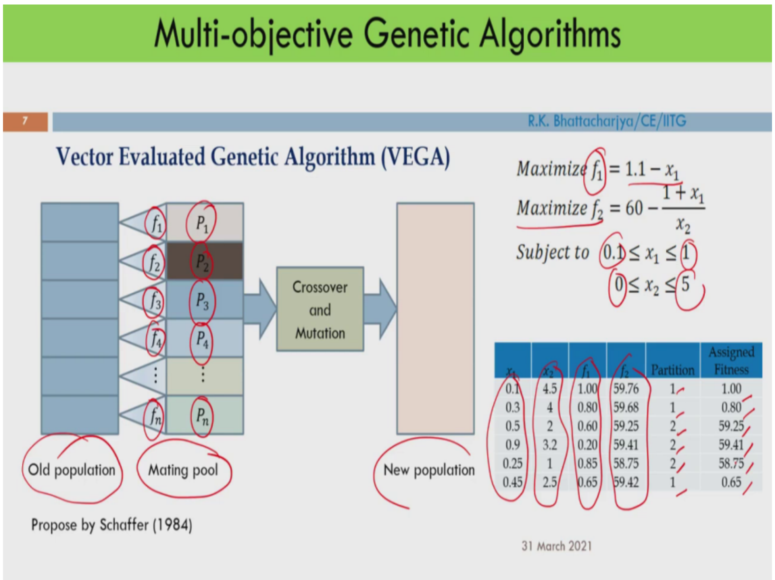
So, let us see, suppose I have this old population and from the old population, I am getting the mating pool. One way to do that, this method is known as Vector Evaluated Genetic Algorithm, VEGA we call it. In this case suppose we have total n objective, then you just divide this population in an subset, ok.

So, this is set 1, this is set 2, this is set 3 this is set 4 and this is set n. So, total n objective. Now, what we will do basically that, that this population 1P1 will be evaluated in terms of objective 1, P2 will be evaluated in terms of objective 2, P3 will be evaluated interms of objective 3, P4 will be of in terms of objective 4, and P n will be in terms of objective n.

Then basically, so after that; so you can apply your crossover and mutation operator and you can create your new population. So, now, what will happen? P1solutions are better in first objective. So, therefore, you will get P1 solution somewhere here, this is actually you are trying to maximize f 1.

Similarly, suppose P2 solution is better in second objective. So, you are getting your population somewhere here; because you have evaluated this population in terms of objective 2 and this is in terms of objective 1. So, therefore, you are getting your solution either at maximize that maximize your f 1. So, which will try to maximize f 1 and which will try to maximize f 2. So, you will not get any solution in between.

(Refer Slide Time: 16:59)



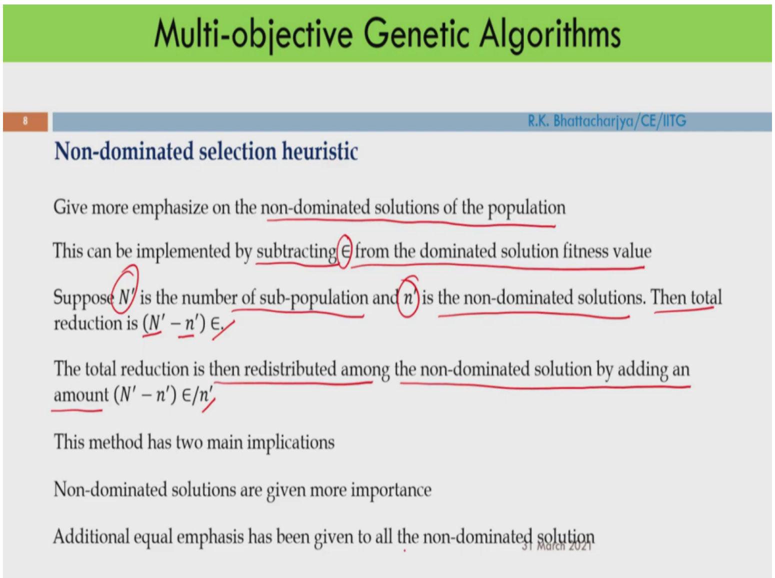
I have shown the example problem here that is maximize f 1, which is 1. 1 minus x 1 and similarly maximize f 2 and it is between 0; x 1 is between 0. 1 and 1, and x 2 is between 0 and 5. So, I have taken some sample solution, suppose these are the solution randomly generated or this is x 1 and similarly these are the solution x 2; this is your objective function 1 that is f 1 and I am also calculating f 2.

And now I have a partition; that means solution 1, 2 and the last solution 6th solution is evaluated in terms of objective function 1. So, therefore, their fitness value is 1, 0.8 and then 0.65. And similarly the solution third, fourth and fifth are evaluated in terms of objective 2. So, therefore, their fitness value is 59.25, 59.41, and 58.75.

So, now you apply the selection operator. So, in that particular group and then you apply the crossover and mutation operator. So, here after few runs what will happen? So, you will get the best solution in terms of objective 1 and you will also get the best solution in terms of objective 2; but this is not a very good method, because you are not getting the entire pareto optimal front.

So, you are only getting the solution at the extremes extreme points; that means that you are maximizing f 1 and you are maximizing f 2. So, you are not getting solution in between. Now, let us discuss non dominated selection heuristic.

(Refer Slide Time: 18:43)



In this case we are giving more importance to the non-dominated solution of the population. So, we are emphasizing the non-dominated solution of the population. This can be implemented by subtracting epsilon from the dominated solution fitness.

That means I would like to reduce the fitness value of the dominated solution and I would like to increase the fitness value of the non-dominated solution. So, what I am doing; so I am reducing the fitness value by epsilon from the dominated solution and then this fitness are added equally to the non-dominated solution.

Suppose N dash capital N dash is the number of sub population and small n dash is the non-dominated solution. So, in the sub population you have small n dash is the number of non dominated solution. Then total dominated solution is N dash minus small n dash and therefore, the total reduction is N dash minus small n dash into epsilon.

So, now we are equally adding this to the non-dominated solution. The total reduction is then distributed among the non-dominated solution by adding an amount which is equal to n dash capital N dash minus small n dash into epsilon divided by small n dash.

So, this method has two main implication. So, what is that, that non dominated solutions are given more importance; that means we are trying to give more importance to the non-dominated solution. And we are also giving additional equal emphasis has been given to all the non-dominated solution. So, as I have explained that, non-dominated solution you cannot tell which one is better and which one is inferior. So, therefore, we are giving equal importance to the non-dominated solution.

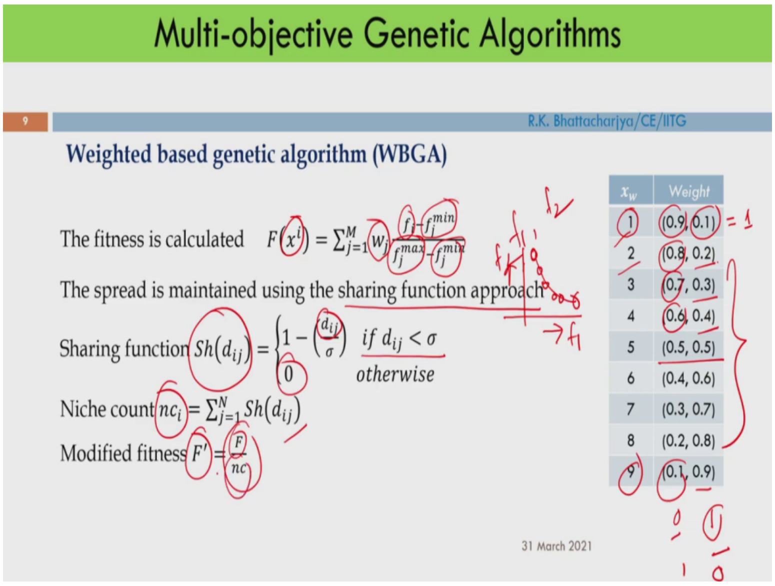
So, in this method is very simple. So, what we are doing? We are trying to reduce the fitness value of the dominated solution and this fitness value we are adding to the non-dominated solution. How we are adding? Equally; so we are giving equal importance to all non-dominated solution.

In that way what will happen; when it will go through the selection operator, selection operator will discard the dominated solution, because you have reduced the fitness value of the dominated solution. And you are you have added or you have increased the fitness value of non-dominated solution.

So, therefore, once the solutions are passing through the selection operator; selection operator will discard the dominated solution and it will actually, it will select the non-dominated solution. So, this is the basic idea of this particular algorithm.

Now, as I said earlier. So, when we are dividing the population in terms of objective. So, we are getting the solution at two extremes, ok. So, either maximization of f 1 or maximization of f 2 or minimization f 1 or minimization of f 1, so we are getting the solution at two extreme. So, we are not getting solution in between. So, what we can do basically? So, we can also means give weightage to each of the solution in order to get other solution.

(Refer Slide Time: 22:15)



So, let us see this is, this method is known as Weighted Based Genetic Algorithm WBGA. So, here fitness value of a particular solution x i is calculated using this equation; that is your we are giving weightage to that particular objective function. So, I will also explain that one;

that is f i minus f j minimum ok and then f j maximum minus f j minimum, this is for a particular objective.

Now, suppose you have two objective, that is your f 1 and f 2. So, what are the weight we are giving? So, if you are giving for a particular solution; suppose this is your x 1 and we have total 9 solution and for solution 1, we are giving weightage to f 1 is 0.9 and weightage to f 2 is 0.1.

Similarly, for solution 2, we are giving 0.8 and 0.2; this is 0.7 and 0.3, 0.6 and 0.4 so on and last solution we will have 0.1 and 0.9. So, these weights are fixed in such a way that, the summation of this equal to 1; that means summation of weight is equal to 1.

So, in that case what will happen? So, we will we are also trying to, we are also trying to get the other point. Suppose if I say that is 0 and 1; that means I am giving your maximum weightage to objective 2, that means I should get this particular solution this is your f 2 and this is your f 1.

So, if I give 0, 1; that means I am giving 0 weightage to objective function 1 and 1 to objective function 2. So, in that case I should get this solution. Similarly, if I give 1, 0; so in that case I should get the this solution. And if I take in between, then I will get the others I will get the other solution.

So, that way depending upon how many solution you want on the pareto front, so you can choose the number of solution and that way you can define the weight. So, in this case what will happen? You are not only getting the extreme values; that means you are maximizing f 1 or maximizing f 2, you are also getting the other solution.

In the last method that, VEGA we have discussed. So, what we have done? Suppose there are two objectives. So, if you have two objectives; that means you have divided your population in two sets, ok. So, some solution will be evaluated based on objective 1 and some solution will be evaluated based on objective 2; that means how much weight you have given to objective 1 and how much objective 2.

So, when you are evaluating the population in terms of objective 1; that means you have given 0 weightage to objective 2. Similarly, when you are evaluating the population in terms of objective 2, you have given 0 weightage to objective 1 and 1 weightage to objective 2.

So, therefore, you are getting two extreme. But in this case what is happening? We are also evaluating the population in terms of other weights, the way we have this we have shown here. So, if I can give 0.9 and 0.1, 0. 8 and 0.2; so in that case or 0.5 and 0.5. So, you are also getting the other points on the pareto optimal front.

Now, the next step is how to maintain the spread; that means I do not want that the solutions are crowded at a particular location, that means I am getting many solution at a particular location. So, I would like to maintain a distance between this solution.

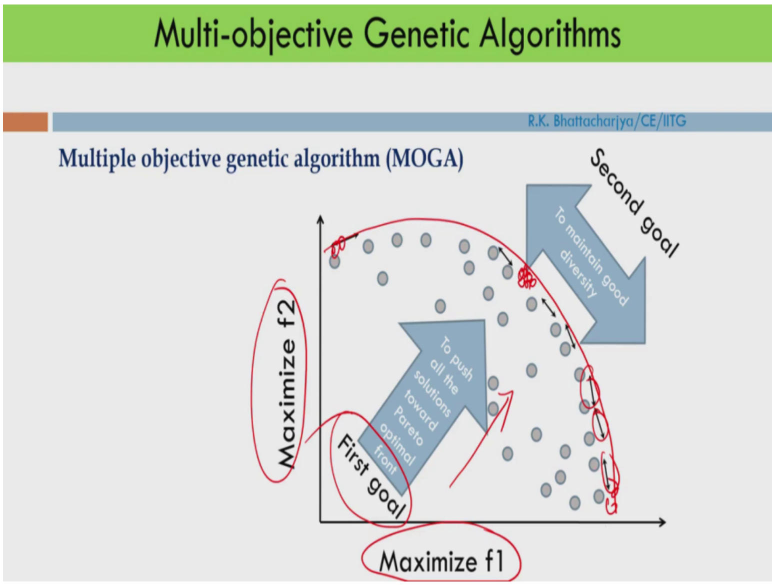
So, we can use the crowding distance criteria. So, in this case, the spread is maintained using sharing function approach. So, already we have discussed sharing function approach. So, I have to calculate d i j and the sharing function I can calculate. So, which is 1 minus d i j by sigma; if d i j is less than sigma, otherwise it is 0. So, if di j is less than sigma; that means this is a crowded solution and otherwise it is a non-crowded solution.

So, what you can do basically, I can calculate what is niche count, ok. Now, what niche count is giving, already we have explained that one. So, niche count is giving that, if niche count value is very high; that means the solution is a crowded solution. If niche count value is not very high or it is 1, in that case I can say that solution is not a crowded solution. So, based on that niche count value, I can tell whether a particular solution is a crowded solution or non-crowded solution.

Now, it now in this case we have calculated modified fitness; because I want to avoid the crowded solution. So, therefore, the modified fitness is calculated, which is original fitness capital F by n c; idea is that, you reduce the fitness value of the crowded solution. So, that is the idea. So, you calculate niche count and once you are getting the niche count; then you divide fitness value by niche count in order to reduce the fitness value. So, idea is that, you reduce the fitness value of the crowded solution.

Now, what is the disadvantage? Again as I discussed already, the main disadvantage is that this is a parameter based method; that means what you have to do, you have to define a suitable value of sigma, otherwise you may not get the favourable solution.

(Refer Slide Time: 27:57)

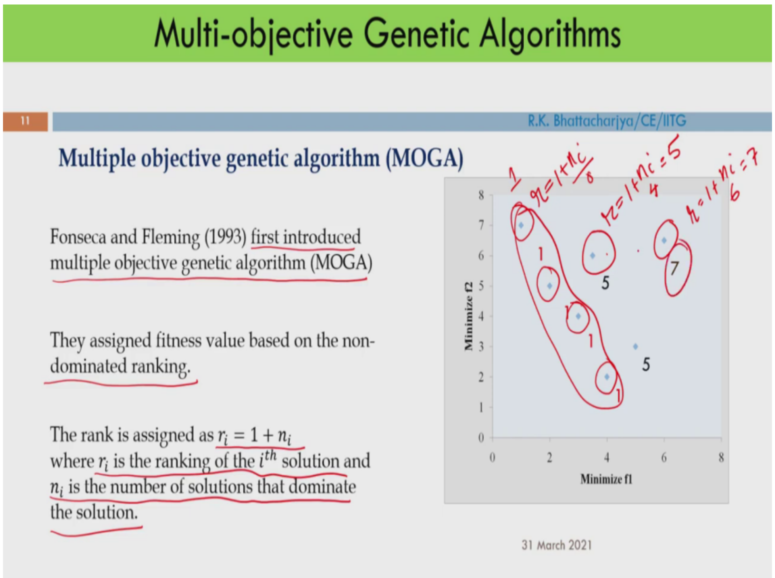


Now, with this background let us discuss the Multi Objective Genetic Algorithm, we call it MOGA. So, MOGA has two objectives; as I have shown you, suppose in this case we are maximizing f 1 and maximizing f 2.

So, here there are two goals, suppose if you want to design an algorithm, so this algorithm will have two goals; the first goal is that you are trying to push the solution towards the pareto optimal front. So, pareto optimal front is somewhere here. So, you want to push your population towards the pareto optimal front.

So, this is the first objective of the algorithm and the second objective is that, I would like to maintain a good spread between this solution; I do not want that my solution is somewhere here, all solutions are crowded either at the extreme points or at some other location. I want basically good spread between the solution. So, these are the two objective of this multi objective genetic algorithm. So, let us discuss multi objective genetic algorithm MOGA.

(Refer Slide Time: 29:00)



Fonseca and Fleming in 1983, first introduced multi objective genetic algorithm. So, they assign fitness value based on the non-dominated ranking. So, what they have done? So, they use a ranking value. So, that means, they assign a ranking value based on the non-dominated concept.

How they have assigned ranking? So, the rank is assigned as r i, which is equal to 1 plus n i, where ri is the ranking of ith solution and ni is the number of solution that dominate the solution.

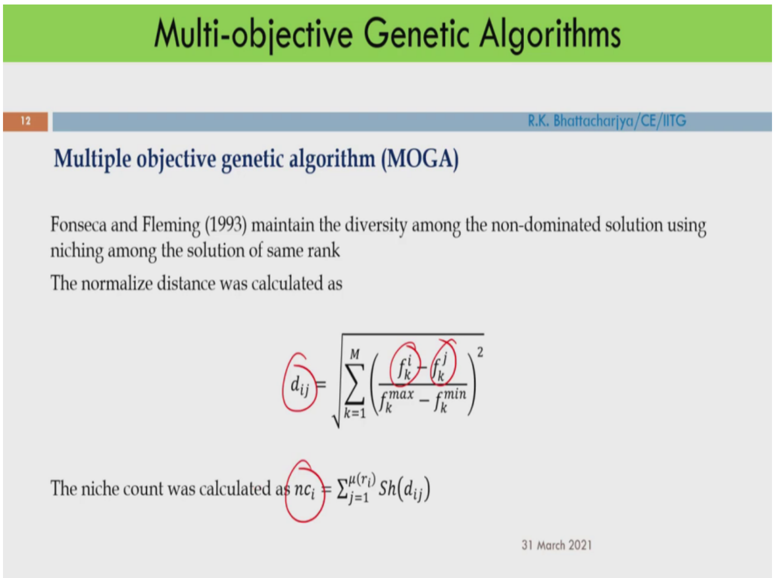
So, let us see this example problem and if you look at this solution, they are not dominated, they are non-dominated solution; that means this solution is not dominated by any other solution, this solution is also not dominated by any other solution, this is also not dominated, this is also not dominated, therefore these solutions are non-dominated solution.

So, if I calculate the rank. So, rank of this particular solution is I plus n i; that mean show many solution are how many solutions are dominating this particular solution. So, n i is 0; so therefore, rank of this particular solution is 1. So, rank is this solution is 1, this is also 1, this is also 1 and this is also 1, all of all this solution will get rank 1.

Now, let us go to this particular solution. This is the dominated solution. Who is dominating? All these four solutions are dominating this particular solution based on non-domination concept. So, therefore, rank of this particular solution is 1 plus n i. So, n i in this case 4. So, therefore, rank should be equal to 5.

So, rank is equal to 5. Now, let us go to this particular solution. So, this solution is dominated by all these solution; therefore rank of this solution is 1 plus n i and in this case n i is 6, so therefore, rank will be equal to 7. So, we are getting 7 ranking. So, now we are getting the ranking, this ranking will be used during selection of this solution.

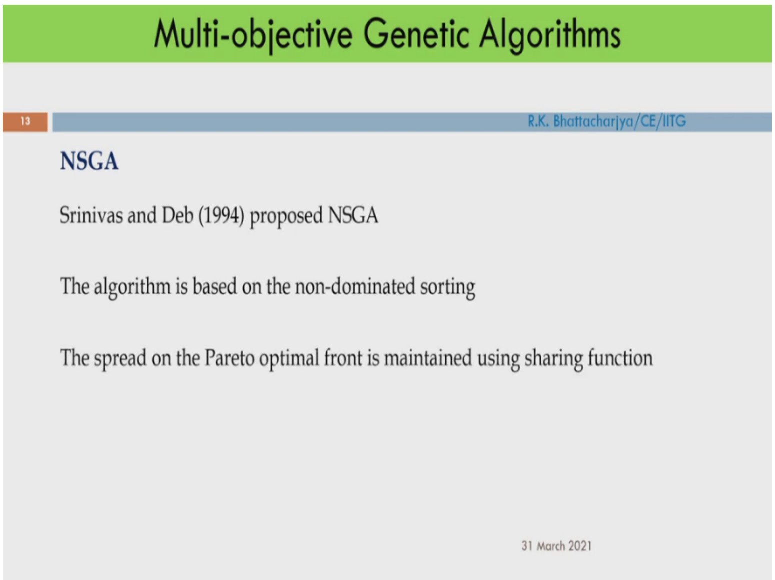
(Refer Slide Time: 31:15)



Fonseca and Fleming maintain the diversity among the non-dominated solution using niching among the solution of same rank. Now, you need to maintain the diversity between the solution of the same rank. So, they have used the niching criteria. The normalize distance was calculated as; so they calculate the distance on the objective function, not on the solution space, ok.

So, as I said, so we have two search space; one is the decision variable space and one is the objectives function space. So, this distance we are maintaining on the objective function space, not at the decision variable space. So, therefore, distance you are calculating between the solution. So, we have calculate the distance of a particular solution in the objective function space. And then we are calculating the niche count and based on that, we are actually doing the selection operation.

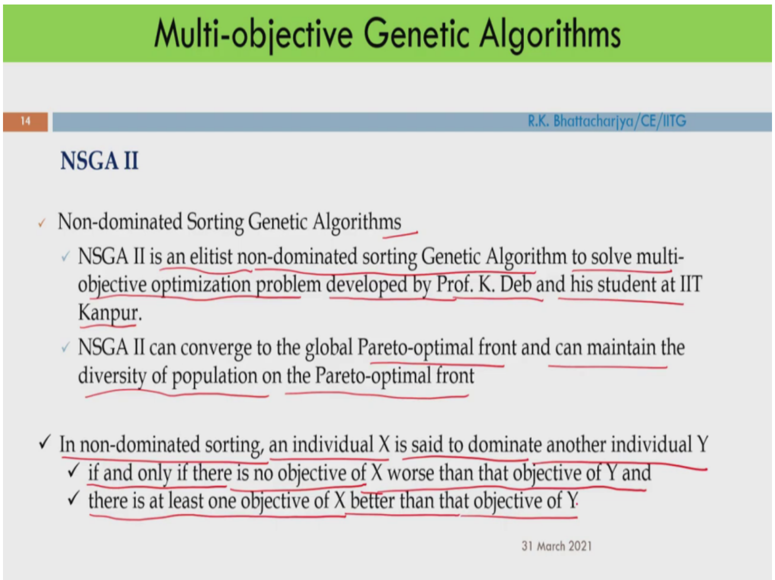
(Refer Slide Time: 32:21)



The next algorithm is NSGA, that is Non-dominated Sorting Genetic Algorithm. This algorithm was proposed by Srinivas and Deb in 1994. The algorithm is based on the non-dominated sorting. The spread on the Pareto optimal front is maintained using sharing function. .

So, what they have done; the algorithm is based on the non-dominated concept and the spread was maintained using sharing function value. As I said, when you are using sharing function value; so it is a parameter based algorithm, that means you have to define the value of sigma. So, that is the main disadvantage of this algorithm.

(Refer Slide Time: 33:00)



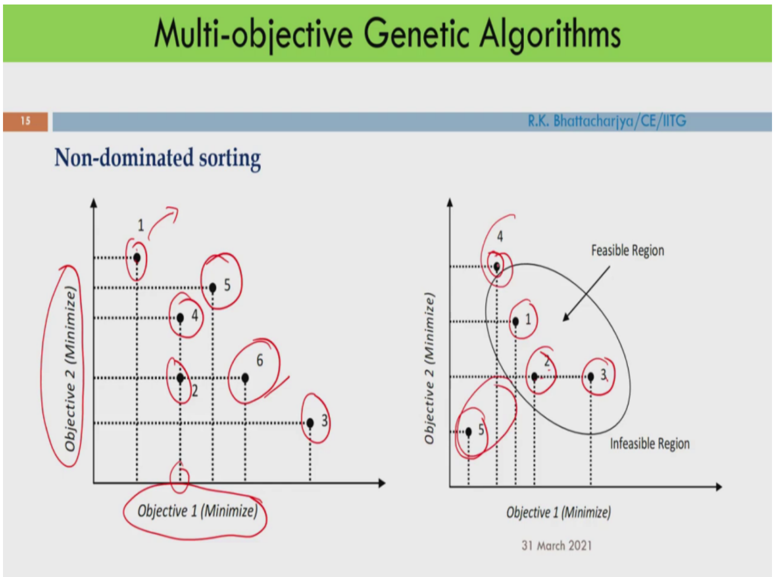
The next algorithm is NSGAII. This is the modified version of NSGA algorithm. So, in this case the spreading is maintained using crowding distance and crowding distance is calculated using a parameter less algorithm.

So, here NSGAII is non-dominated sorting genetic algorithm. So, NSGAII is a elitist non-dominated sorting genetic algorithm to solve multi objective optimization problem and this algorithm was developed by professor Kalan Deb and his student at IIT Kanpur. NSGAII can converge to the global Pareto optimal front and can maintain the diversity of the population on the Pareto optimal front.

So, these are the two main advantages of NSGAII. So, it can reach very quickly the Pareto optimal front and it can also maintain the diversity on the Pareto optimal front, spread can be maintained.

Now, what is non-dominated sorting? In non-dominated sorting an individual X is said to dominate another individual Y, if and only if there is no objective of X worse than the objective of Y and there is at least one objective of X better than the objective of Y, ok. So, if these two criteria are satisfied, then you can tell that the solution, the both the solutions are non-dominated solution.

(Refer Slide Time: 34:40)



Let us see an example problem. Suppose we have solution 1, solution 2 and solution 3. If I look at solution 1, solution 2 and solution 3; these solutions are non-dominated solution. Let us compare solution 1 and solution 2. Solution 1 is better in objective 1 because this is a minimization problem, but it is not better or inferior in objective 2.

Similarly, solution 2 is better in objective 2, but not better in objective 1. So, therefore, these two solutions are non-dominated solution by the definition we have given in the previous slide. Similarly, if I compare solution 2 and 3, both are non-dominated solution.

Now, let us see the solution 4 and solution 2. So, in this case the solution 2 is better than solution 4 in objective 2 and solution 2 is not inferior than solution 4 in objective 1. So, therefore, solution 4 is dominated by solution 2 or you can say that solution 4 is a dominated solution.

Similarly, if I compare solution 6 and solution 2, the solution 2 is better than solution 6 in objective 1; but in case of objective 2, solution 2 is non-inferior than solution 6. So, therefore, solution 6 is a dominated solution and solution 2 is not a dominated solution.

And similarly if I compare solution 5 and solution 4, solution 5 is dominated by solution 4 in both the objective. So, therefore, they are this is your dominated solution; but if I say solution 1, solution 2 and solution 3 they are non-dominated solution.

Now, if you have a constraint. So, I say that this is the feasible region; that means solution 1 and solution 2 and solution 3 are feasible region and they are dominated, either dominated or non-dominated solution. Suppose if I compare solution 1 and solution 2, solution 1 and solution 2 are non-dominated solution; but if I compare solution 2 and solution 3, solution 3 is a dominated solution ok, solution 3 is a dominated solution.

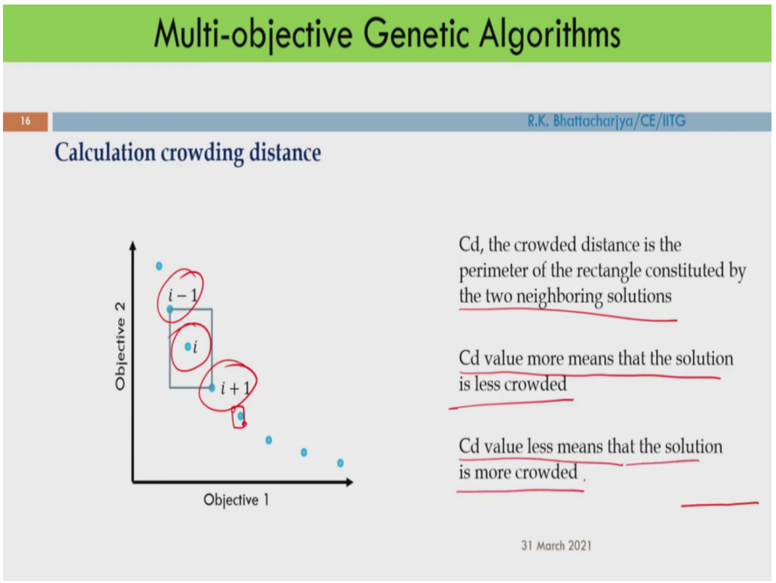
Now, let us see how we can compare this solution. Suppose if I compare here solution 1 and solution 2, I can say that they are non-dominated solution and one of them will go to the next generation. Or if there is a tournament between one and 2; so in terms of ranking would will have the same ranking and therefore, they cannot be compared in terms of ranking. But if you are comparing solution 1 and solution 5; you can say that solution one is better than solution 5 and solution 1 will go to the next generation.

Here if I compare solution 1 and solution 2. So, both the solution will have same rank. So, therefore, we cannot tell which will win the tournament or and who will basically go to the next generation; but when solution 2 is playing with solution 3, solution 2 is better than solution 3, therefore solution 2 will go to the next generation.

Now, let us compare solution 4 and solution 2. So, solution 4 is in the infeasible region and solution 1 in the feasible region. So, therefore, if there is a tournament between a infeasible solution and feasible solution; solution 1, feasible solution will win the tournament.

Now, let us see the solution 5 and solution 4. So, both are at the infeasible region. So, in that case what will happen; but the constant violation of solution 4 is lesser than constant violation of solution 5. So, therefore, solution 4 will win the tournament. So, I can apply these basic rules and to select that solution, so based on the ranking as well as crowding distance. So, let us discuss what is crowding distance.

(Refer Slide Time: 38:52)

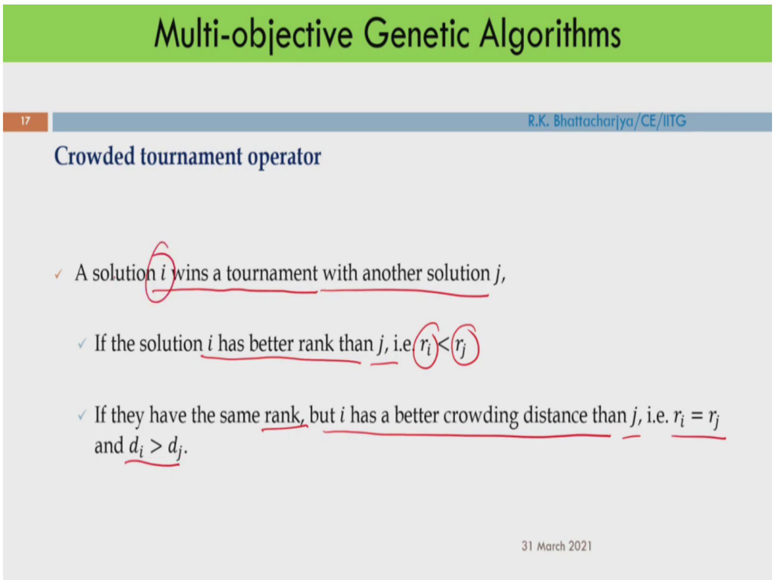


So, crowding distance is, in this case the crowding distance is calculated; suppose I would like to calculate the crowding distancefor ith solution. So, what I will do; I will take two solution nearby solution that is i minus 1 and i plus 1 and then I will calculate the perimeter of this rectangle. If the perimeter value is large, in that case the solution is not a crowded solution; if perimeter value is small, you can tell the solution is a crowded solution, ok.

So, suppose there is a solution somewhere here; suppose I have a solution somewhere here, another solution here, then this will be the perimeter of this particular your solution. So, therefore, if the perimeter value is less, so in that case I can tell the solution is a crowded solution; if the perimeter is large, in that case I can tell the solution is not crowded solution.

So, now I will have two things, one is the ranking, that is the non-dominated ranking and another one is the crowding distance. Now, based on that so, as I said C d, the crowding distance is the perimeter of the rectangle constituted by two neighbouring solution. C d value more means the solution is less crowded and C d value less means, C d value less means that the solution is more crowded.

(Refer Slide Time: 40:22)

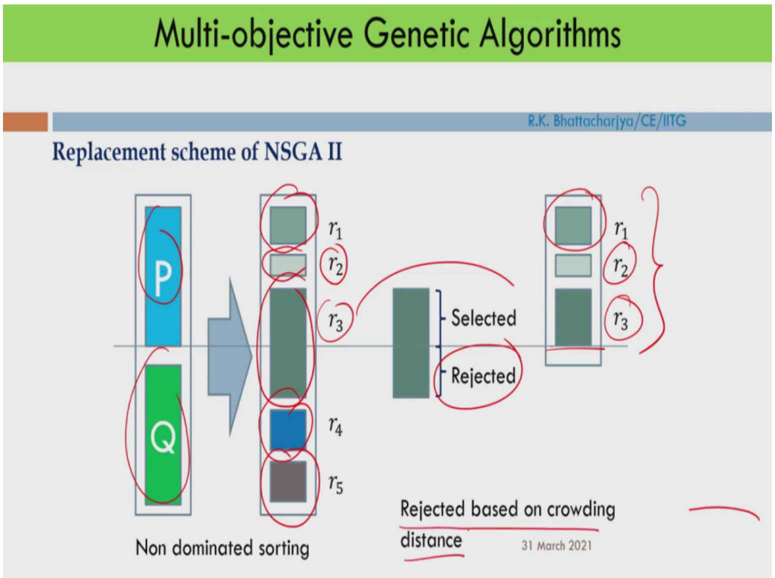


Now, as I said that we have to apply the selection operator based on the ranking as well as crowding distance. So, how we are doing that? A solution i will win the tournament with another solution j; if the solution i has better rank than j, that means ri is less than rj. That means the first non-dominated solution will get the rank 1 and then the second non dominated solution will get the rank 2.

So, therefore, rank lesser rank means better solution. If a solution i has a better rank than solution j, then the solution i will win the tournament. If they have same rank, but i has a better crowding distance than j; that means r i equal to r j and d i is greater than d j, in that case i will win the tournament.

So, what we are doing here, first we are looking at the ranking; if they are different rank, the lesser rank will be selected. If rank is same, then we will look at the crowding distance; then we will select the solution with more crowding distance. So, more crowding distance means; that means the solution is less crowded.

(Refer Slide Time: 41:38)



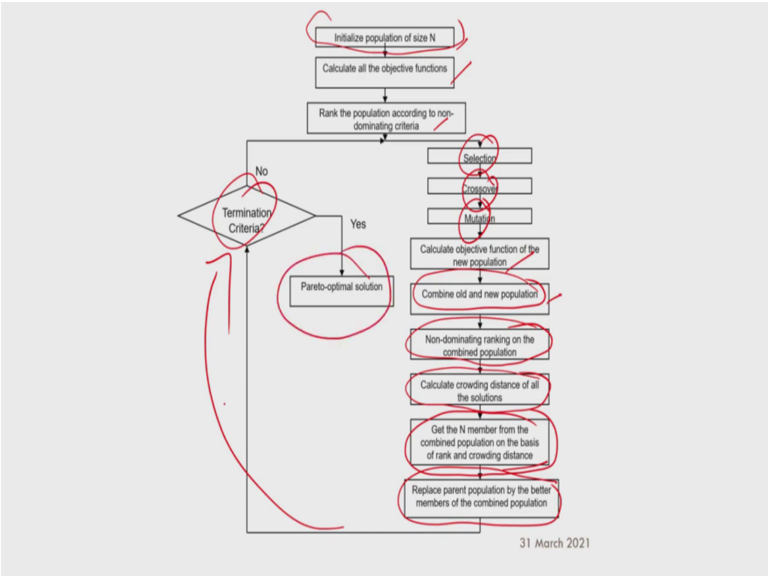
Now, let us see. So, this is the population and we have created a new population Q then we are combining this population and we are doing the sorting, non-dominated sorting. And these are the solution with rank 1, they are the non-dominated solution. And after that if you are removing that, if you go for non-dominated sorting. .

So, this is the rank 2 solution, then this is the rank 3 solution, this is the rank 4 and this is the rank 5 solution. So, what we are doing; so we are initially combining this population, the old population and new population P and Q. By combining what we are doing, we are also implementing elitism, ok.

And then we are doing the ranking of this combined solution. So, rank 1, rank 2, rank 3 rank 4 and rank 5. So, in this case, so this is the non-dominated sorting. And after that, so we are selecting all rank 1 and then we are selecting suppose rank 2 and then suppose we cannot actually take all the rank 3 solution, because we there is no space. So, the solution will be selected based on the crowding distance, ok.

So, the more crowd, the solution with more crowding distance will go to the next generation and I am getting the solution for the next generation. And this solution will be rejected based on the crowding distance so, as I said rejected based on the crowding distance.

(Refer Slide Time: 43:09)



Now, this is the algorithm of NSGAII. So, initialize population, this is similar to genetic algorithm; calculate objective function value, similar to simple genetic algorithm, then we are giving the ranking based on non-dominated criteria.

Now, you apply selection, crossover, mutation, calculate the objective function, then combine the population; by combining, we are actually doing elitism here in this particular steps. And then we are doing the non-dominated sorting and based on that, we are putting the rank and we are also calculating the crowding distance.

And then N member of the population is selected based on ranking and crowding distance. So, replace the parent population by the better members of the combined population and this process will continue.

So, unless you are not reaching the termination criteria, so one you are reaching the termination criteria; then you will come out of this loop and you will get the Pareto optimal front. So, this is all about NSGAII algorithm; this is one of the very efficient algorithm for solving multi objective optimization problem. So, let us stop here.

Thank you; see you in the next class.