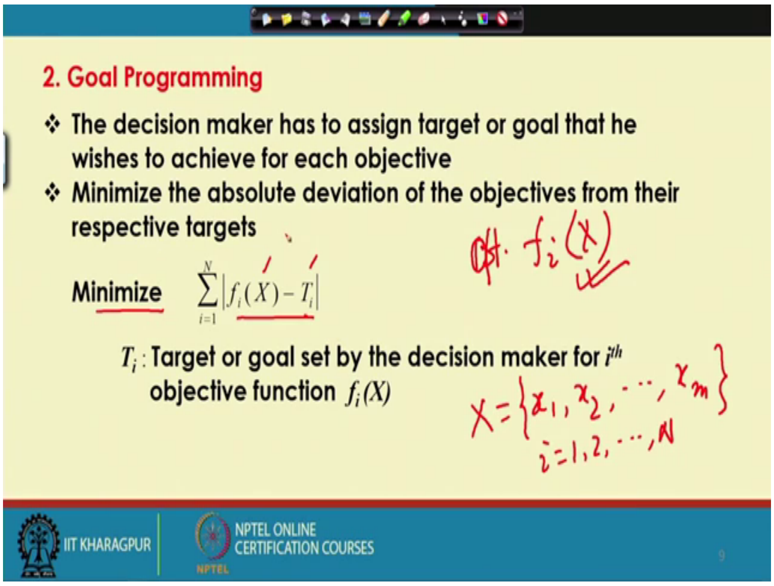
Traditional and Non-Traditional Optimization Tools
Prof. D. K. Pratihar
Department of Mechanical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 23
Multi-Objective Optimization (Contd.)

Let me discuss another approach which is also very popular. And this is once again one of the most popular traditional approaches to solve this type of multi objective optimization problem. Now this approach is known as the goal programming.

(Refer Slide Time: 00:35)



Now, here actually what you do is? The decision maker will have to set the target for each of the objectives which is going to maximize or minimize. Now, supposing that the targets are set. Now the obtained objective function value will be compared with their respective target values and the deviation or the error that would be decided.

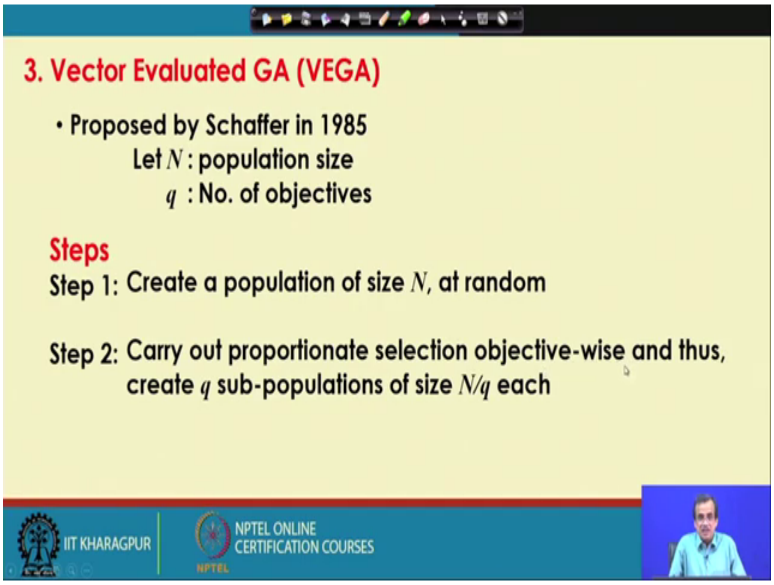
Now, here this deviation or the difference between the calculated value of this particular objective and it is corresponding target value. So, this particular difference it could be either positive or negative and just to get the positive value we consider the mod. So, mod of f i X minus T i is nothing, but is actually the deviation or the error in the mod value that is the positive value and this particular the deviation. The sum of all the deviations considering all the objectives; so we try to minimize. Now here once again the problem is the same supposing that say I want to say maximize or say minimize let me put I want to optimize; it could be either maximize or minimize.

So, optimize f i X and X is nothing, but a collection of all small x values say I have got m number of small x values or the design variables. And what I do is and this particular I that is the ith objective function and I varies from 1, 2 up to capital n. So, I have got capital n number of objectives. Now for each of the objectives; so we try to find out the difference between the calculated objective and the target value; at this particular difference we consider the mod we sum them up and then we try to minimize; that means, we try to minimize the difference between the calculated objective and its target value; that means, your through the optimization.

So, we will try to hit the target value for each of these particular the objective. So, that is the philosophy behind this type of the goal programming. Now here I am written we try to minimize the absolute deviation of the objectives from their respective targets. Now the problem is like how to set this particular the target. Now if I take the same example like your; the cost of a car and the accident rate of this particular car; and we have already discussed that if cost is more the accident rate will be less and vice versa. Now here actually what I do is how to set the target for the objective that is the cost. So, what is the maximum amount beyond which I should not go?

Similarly what is the optimal accident rate which we should not exceed. Now how to set this particular target that could be the problem. Now once this particular targets are set; it is now we can use this goal programming to find out or to solve this type of multi objective optimization problem.

(Refer Slide Time: 04:55)



Now this is actually how to use the principle of goal programming to solve the multi objective optimization problem. Now then comes actually I am just going to discuss a third approach this is actually popularly known as vector evaluated genetic algorithm in short this is known as vega. Now before I start let me tell you that this particular approach is the first approach using genetic algorithm to solve the multi objective optimization problem; and this particular approach was proposed by schaffer in the year 1985.

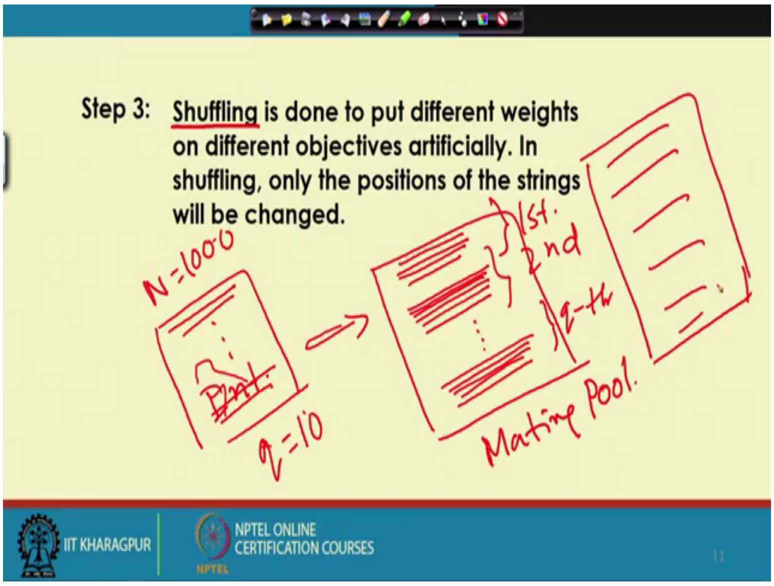
So, as I told this is the first approach. So, let us let us try to find out the principle, but this is not a very good approach very efficient approach, but as this is the first approach using a genetic algorithm. So, let us let us spend some time on discussing the principle of this particular the approach. Now here before I start; now let me let me tell you now here what we do is?

We try to copy the first approach; that is the weighted sum approach in an artificial way inside the genetic algorithm; how to do it I am just going to discuss? Supposing that I have got a multi objective optimization problem, having say q number of objectives. So, q is the actually the total number of objectives. Now to solve this particular problem what you do is we generate a large population size that is denoted by say capital n

now let us see how to proceed with this particular the approach. How does it work? Step one: we create a population of size n at random. Now, step two: carry out proportionate selection objective wise thus create q sub population of size N by q each. Now what does it mean supposing that I have got a multi objective optimization problem having say 10 objectives. Now what I do is. So, we try to start with a large population size say might be the population size n could be equal to say 1000.

So, we start with a population of 1000 solution selected at random and next what I do is we carry out proportionate selection to get the mating pool and while carrying out the proportionate selection we do it objective function wise; that means, out of the 10 objectives first we concentrate on the first objective try to find out one sub population then we concentrate on the second objective try to find out the second sub population at this process I am just going to continue and corresponding to the q th objective. So, I will be getting the q th subpopulation of solution. Now let us see how does it works.

(Refer Slide Time: 08:21)



Now, let me let me do one thing whatever I discuss let me just put it here in the a schematic from sort of thing. Now supposing that I have got a population large population that is capital n and that is equals to 1000; and I have got the number of objectives q equals to 10 this is the initial population this is nothing, but the initial population.

Now this initial population I am just going to get one mating pool from this initial population. Now for this mating pool actually what you do is. So, we try to use the proportionate selection and here the proportionate selection is done objective function y.

Let me concentrate on the first objective first supposing that this is the binary coded thing. So, we are got the large number of solutions here. So, we concentrate on the first objective and try to carry out proportionate selection and we will be getting the first sub population. And this sub population is very good in terms of the first objective function only, but it may not be. So, much good with respect to the other objectives.

Next I concentrate on the second objective function for this second objectives; and try to form one sub population. So, this is good with respect to the second objectives and this process is will continue and for the q th one; q th your objective function. So, I will be getting the q th sub population now let me repeat now this q th sub population is very good in terms of the q th objective function, but it may not be so much good with respect to the other objectives.

So, this is the way actually we form the mating pool using proportionate selection and once I have got this particular mating pool now we go for the shuffling. So, shuffling is done just to put different weights on the different objectives artificially. Now if you remember the first approach that is the weighted sum approach what you do is we put different weights on different objectives like w 1 w 2 and. So, on.

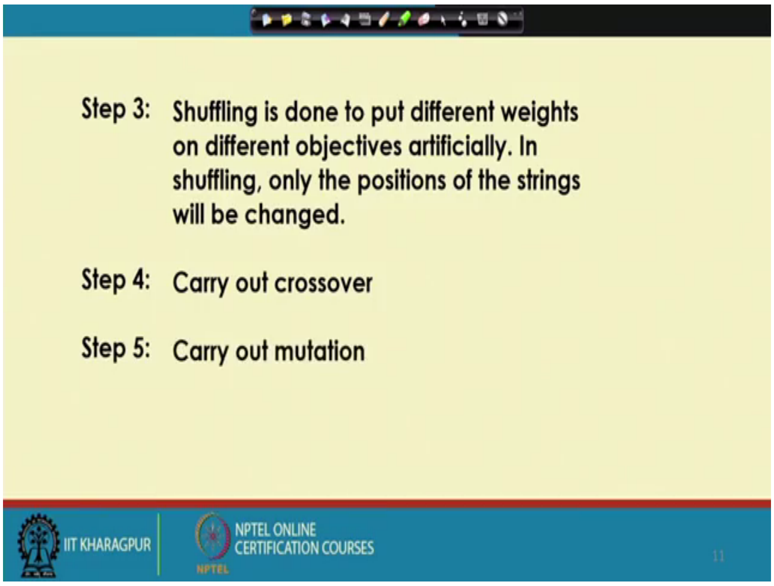
Now, here the moment we do shuffling means what. So, what will do is we will change the position the relative position of the solution just like shuffling of cards. So, the relative position of these particular solutions will be changed. The moment we do the shuffling as if we are putting some artificial weights or the different objectives, but we do not know the actual numerical values of this particular the objectives.

So, this is the way actually we put different weights on the different objectives and it is all most similar to the first approach that is on weighted sum approach and once I have got this particular the shuffled population for example, say here I will be getting this particular the shuffled population.

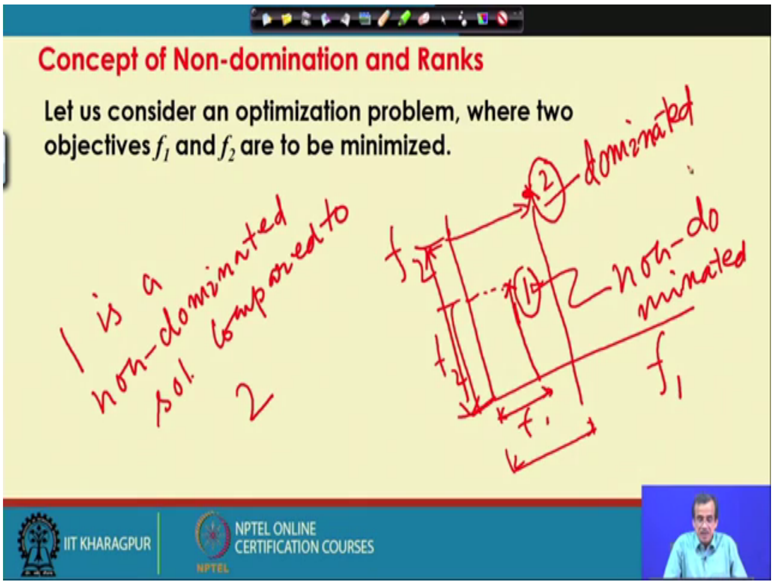
So, this particular shuffled population now it is going to participate in cross over and mutation and that completes actually one iteration or the one generation of this particular the ga. Now this process will go on and go on and through a large number of iteration there is a possibility that we will be getting one population of solution which is very good with respect to this particular the objective function , but as I discuss the concept of pareto optimal front of solution we may not get here we may not be able to draw the pareto optimal front of solution corresponding to this particular the multi objective optimization problem.

So, this is actually a the drawback of this particular approach , but as I told that this is the first approach in which a genetic algorithm has been used to tackle the multi objective optimization problem. So, I thought I should discuss the principle of this particular the first approach using g a , but as I told it has got some drawback in the sense it will not be able to find out what should be the pareto optimal front of solution. So, this particular approach is actually very simple.

(Refer Slide Time: 13:21)



(Refer Slide Time: 13:23)



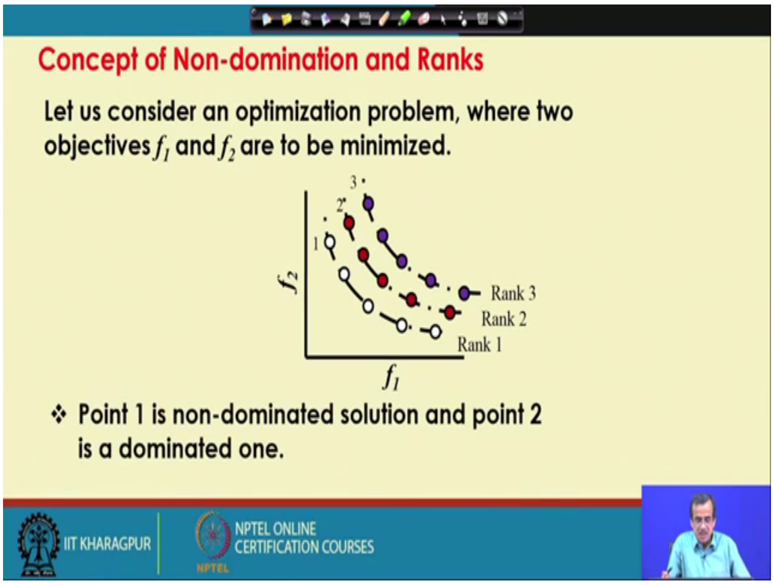
Now, I am just going to start with some more efficient approaches to tackle the multi objective optimization problem using the concept of say genetic algorithm. Now before I go for that. So, I will have to define a particular term that is called the non dominated solution. Now let us see how to define this particular the non dominated solution. Now let us consider an optimization problem having say two objectives.

Now here actually what you do is let me let me just draw say I am just going to consider a two objective optimization problem. So, f 1 and f 2 and our aim is to minimize say both f 1 and f 2. Now let me consider any two solution selected at random let me consider I have got a solution here say one I have got another solution here say two.

Now, if I compare between these two solutions in terms of the objective function now here as I told our aim is to minimize both f 1 and f 2 now corresponding to the first solution. So, this is nothing, but f 2 and this is nothing, but f 1. So, this is f 1 and this is your f f 2 similarly corresponding to this particular solution. So, this is my f 1 and this is my and this is my f 2 this is my f 2. Now if I compare these two solution; solution 1 and solution 2 in terms of f 1 and f 2. So, corresponding to this f 1 is more f 2 is more and it corresponding to this particular solution one f one is less f 2 is less and these are minimization problem.

So, this solution 1 is a better solution compared to solution 2. In terms of both f 1 and f 2; that means, we can declare that solution 1 is non dominated solution is a non dominated solution non dominated solution compared to compared to solution 2. So, solution 1 is a better solution compared to solution 2 and this one is nothing, but non dominated solution which cannot be dominated. So, this is a better solution and solution 2 is nothing, but the dominated solution and which is actually a one solution this is actually the definition of the non dominated solution. So, this particular term we are going to use very frequently while describing the principles of some other tools using g a to solve the multi objective.

(Refer Slide Time: 16:58)

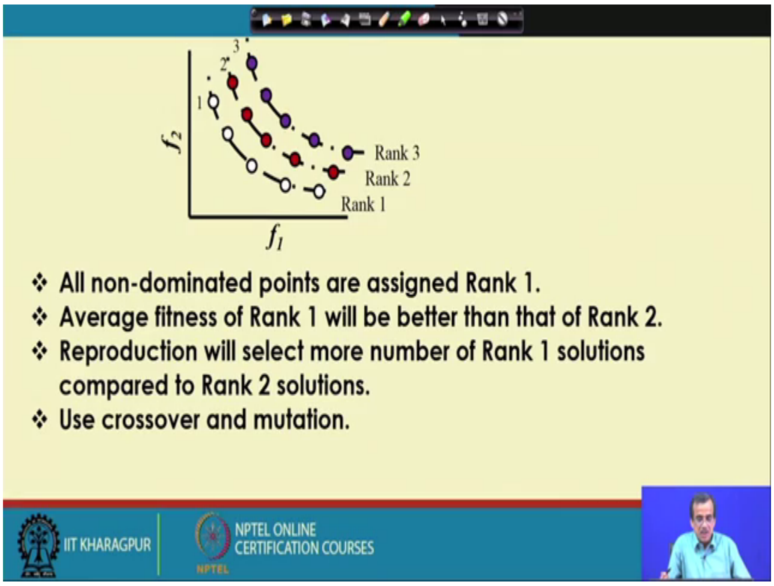


Optimization problem now the same thing I am just showed it here. So, if I compare say this is once again the same problem f 1 and f 2 I will have to minimize. So, if I compare 1 and 2; definitely 1 will be a better solution compared to 2. So, one is nothing, but a non dominated solution. Now similarly if I compare; so this particular solution with that particular solution definitely this will be a better solution or the non dominated solution. And if I compare between this and this will be the non dominated between these and this will be the non dominated between this and this will be the non dominated.

So, so these particular non dominated solutions are nothing, but the rank one solutions we assign a rank one to the non dominated solution. Similarly we have got the rank two solution, rank three solutions something like this and of course, the rank one solutions will be better compared to the rank two solutions. And a rank two solutions will be better compared to the rank three solution.

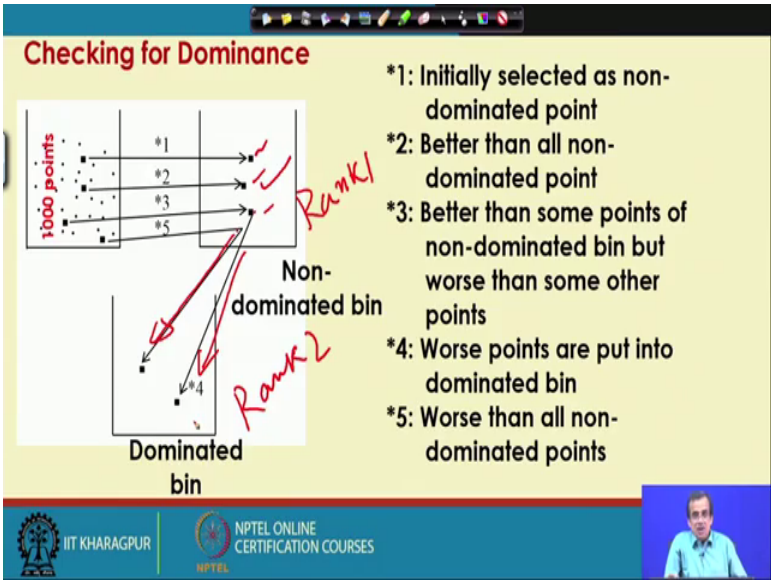
And if I compare the average fitness; so the average fitness of the rank one solution will be better compared to that of the rank two solution. Similarly the average fitness of the rank two solutions will be better compared to that of the rank three solution. Now my question is; how to assign the ranks and how to make a computer program to assign this particular the ranks? Now if I see the literature there exists a few methods like; how to assign this particular the rank? Now here I am just going to discuss say.

(Refer Slide Time: 18:58)



One method like how to assign this particular the rank and how to how to get this rank one solution and the rank two solution.

(Refer Slide Time: 19:08)



Now, once again the problem is as follows say in a in a initial population say I have got a large number of selections selected at random. Now I will have to find out the rank one solution, rank two solution, rank three solution and so on. Now for the time being let me assumes that I want to get only rank one and rank two solution. Let us see how to do it.

Now, supposing that I have got a large number of solutions say denoted by capital n and n equals to 1000 here now what I do is? So, I will have to divide into two ranks rank 1 and a rank 2. So, here I have got one bin here or the bucket here this is nothing, but the non dominated bin and this is also known as the rank 1 bin rank 1 bin and here I have put another bin that is called the dominated bin and that is nothing, but the rank 2 bin.

now this initial population; I will have to distribute two the non dominated bin and this dominated bin. Let us see how to do it in computer program? Now what will have to do is. So, initially we select any point out of these 1000 at random and what you do is we just we just put this particular solution we just put this particular solution in the non dominated bin now supposing that. So, this particular solution there is the solution one that is put in the non dominated bin.

Now, what you do is? So, this is directly selected here for this non dominated bin and it is put there ok. Next we go for the second solution. Now once again this second solution that is selected at random out of the remaining like your nine hundred ninety nine points. So, out of these 999 points say any one is selected at the second point at the second point is brought here to the non dominated bin for the purpose of comparison. Now if it is found to be better than the non dominated points which are lying here already it is put in this particular the non dominated bin. So, the solution two is put here in the non dominated bin then we go for the third point supposing that the third point is such. So, this particular point is such that it is better than some of the points lying on the non dominated bin , but it is found to be worse compared to some other points. So, if it is in between.

So, what I do is. So, that particular third point we put it here and the points which are found to be worse compared to. So, this particular non dominated solution those are forced to the dominated bin. So, these particular the what solution are actually forced to put in the dominated bin now there could be another possibility that a particular solution selected from the initial bin it could be worse compared to all the existing non dominated solution here and if that is the case then that particular solution will be put in the dominated bin and this particular process will have to repeat till we just consider we consider all 1000 points

Now, these particular 1000 points will be divided into or distributed into two bins that is the rank one bin. So, this is nothing, but the rank one bin and this is the rank two bin. So, this will be distributed in to rank one and rank two now if I want to get rank three bin what will have to do is the rank two solution you put it here and repeat the process and if you put the rank two solutions here.

So this rank two solutions here and if you repeat this particular process; so, I will be getting the modified rank two here and the rank three here. So, the rank two solutions will be modified let me take a numerical example supposing that I have got a 1000 points here and say of the rank one basin I have got 200. So, the remaining 800 should come here now I start this particular process with 800 here.

So, here there will be 800 point and will be going to find out the modified rank two and the rank three supposing that if the modified rank two I have got 300. So, out of 800, 300; I have put in the modified rank two. So, the remaining 500 will go to the rank three. So, this is the way the whole population will be divided into rank one rank two and rank three fronts and once I have got this rank one rank two and rank three front. Now actually we can take the help of the genetic algorithm to find out that particular the pareto optimal front of solution now the way the g a will help is as follows like if I have got this rank one rank two and rank three solution. So, what you do is as we have already mentioned that the average fitness of the rank one will be better compared to the average fitness of rank two and average fitness of rank two will be better compared to the average fitness of rank three.

Now, actually the reproduction operator will try to select more number of rank one solution because their average fitness is better. So, there is a possibility that in the mating pool I will have got more number of rank one solution and if we have got more number of rank one solution then it will participate in cross over and mutation there will be diversification of the solution and there is a possibility that I will be getting better and better solution. So, there is a possibility that if I get rank one solution in first iteration it is if it is like this there is a possibility in second iteration I will be getting a pareto front like this third iteration I will be getting a pareto front like this and as iteration proceeds might be after a few iteration.

I will be getting this type of pareto front and g a will try to it write this type of pareto front through a large number of iteration. So, this is the way actually we will have to find out the pareto optimal front of solution now the closer this particular front to this particular the origin the better will be that particular the front in terms of the quality, but we cannot hit this particular the origin that is 0, 0 that is the globally minimum or both f 1 and f 2 will be 0 that cannot be reached.

Because there will be some functional constraint there will be some variable bounds. So, we will not be able to hit you will not be a getting a car at 0 cost at zero accident rate that is very hypothetical, but we will be getting a pareto optimal front of solution which is very close from this particular the origin and that will be the ideal the pareto optimal front of solutions.

Thank you.