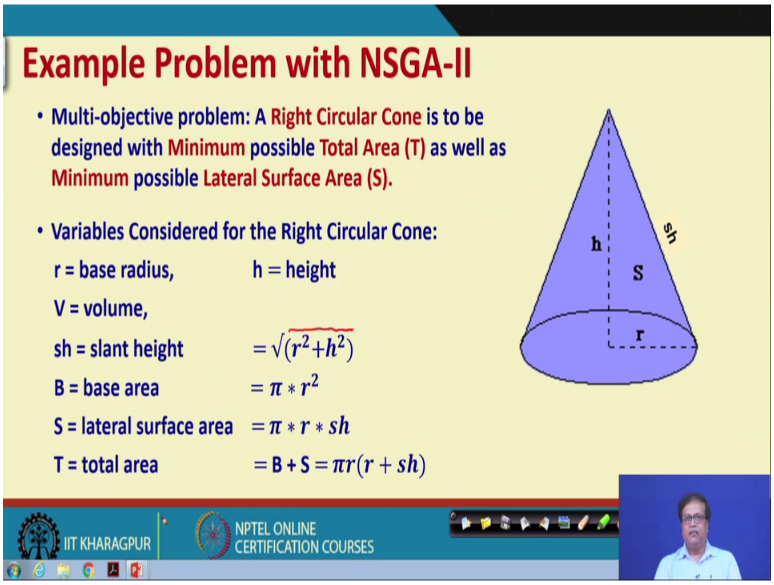
Selected Topics in Decision Modeling
Prof. Biswajit Mahanty
Department of Industrial and Systems Engineering
Indian Institute of Technology, Kharagpur

Lecture – 40
NSGA – II Examples

So, in our subject Selected Topics in Decision Modeling, we are now in our 40th lecture that is NSGA II examples.

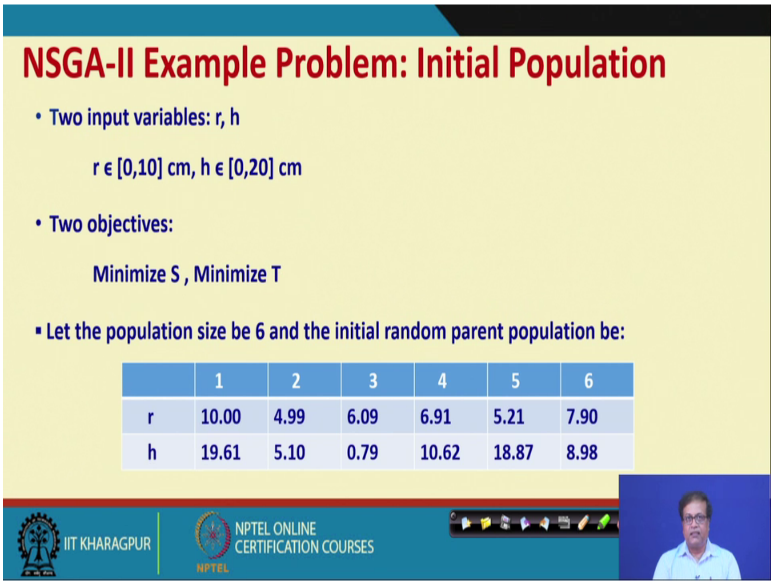
(Refer Slide Time: 00:27)



So, in our previous lecture we have seen how to carry out multi objective optimization, with regard to evolutionary computing meta heuristic technique. And, we have taken up one such technique called NSGA II and, I have discussed the general ideas of NSGA II. So, today's lecture we shall understand that with examples. So, let us take a very ordinary and simple problem, supposing there is a right circular cone and this right circular cone is you know we have to design with minimum possible total area and minimum possible lateral surface area; obviously, with regard to certain constants of the height and other things, height and the radius.

So, these other things r is the base radius, h is the height, V is the volume sh is the slant height which is given by root over r square plus h square, then B is the base area which is pi r square it is a base area, then S is a lateral surface area, that is pi r into sh, sh and pi r and T is the total area that is B plus S so, pi r r plus sh is it alright. So, our job is to really design means taking an r and taking an h that will minimize S and minimize T is it alright. So, that is our example problem.

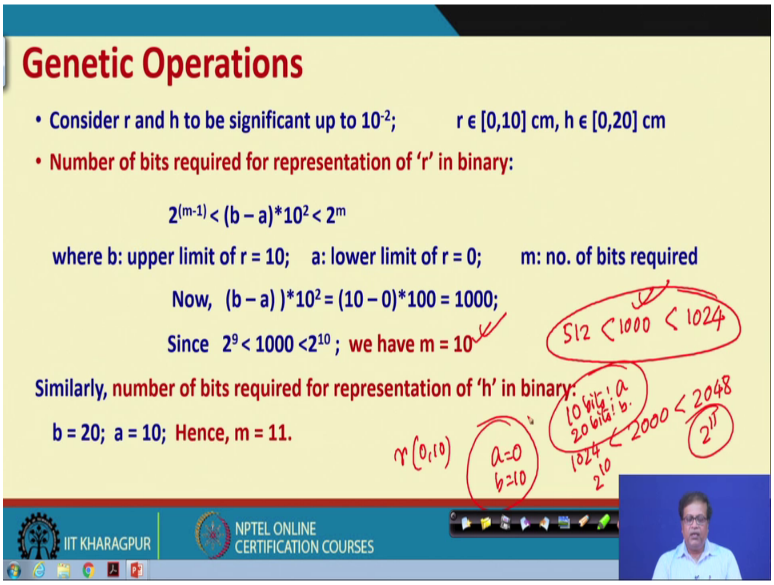
(Refer Slide Time: 02:28)



So, very simple problem really, but then let us see we have taken a non trivial, I mean rather trivial kind of input values that is r is between 0 and 10 h is between 0 and 20, you may think that if r equal to 0 and h equal to 0 that is the best possible solution you are right. But then let us we have taken the simple problem, essentially to really understand that how NSGA II, tries to do this with regard to the minimization of S and minimization of T it is good, that if we also have an idea about the ultimate optimization solution right.

So, let us take some random population so, we have taken let us say 6 random parent population and, this parent populations are randomly obtained so, 10 and 19.61, 4.99 and 5.1, 6.09 and 0.79 etcetera so, these are our initial random parent population. So, in order to carry these out you know first of all we have to encode this.

(Refer Slide Time: 03:51)

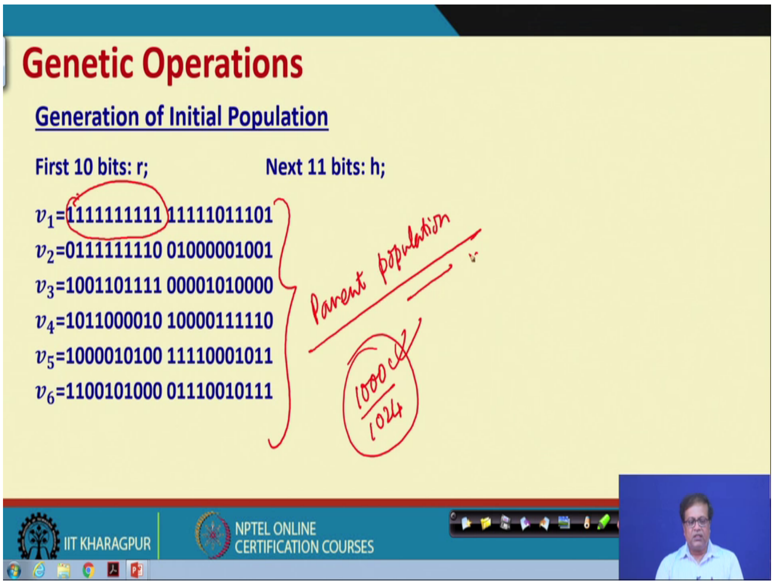


If you recall our ga discussions, then the first thing is to encode. So, the encoding requires r is between 0 to 10 and h is between 0 to 20 so, you can say that b minus a I mean b and a is 0 for r 0, 10 a is 0 and b is 10 right.

So, 2 to the power m minus 1 to 2 to the power m should be b minus a into 10 to the power 2. So, you can see b minus a in this case is 1000 right so, because we need 2 digit accuracy that is why 10 to the power 2. So, supposing we need at least two digit accuracy so, we need to have 1000 points between, you know the binary should be represent at least 1000, points we know that 2 to the power 9 is 512 and 2 to the power 10 is 1024 is not we know that.

So, we know that 5 hundred 12 and 1024 within that 1000 lies. So, to represent 1000 you know, because we want two digit accuracy so, we should represent 1000 different combinations so, we must have m equal to 10 alright. Similarly for h which is between 0 to 20 we need 2000 presentations so, we must have 1024 to 2048 where this one is 2 to the power 10, this one is 2 to the power 11. So, we therefore, we need 10 bits for a and 20 bits for b.

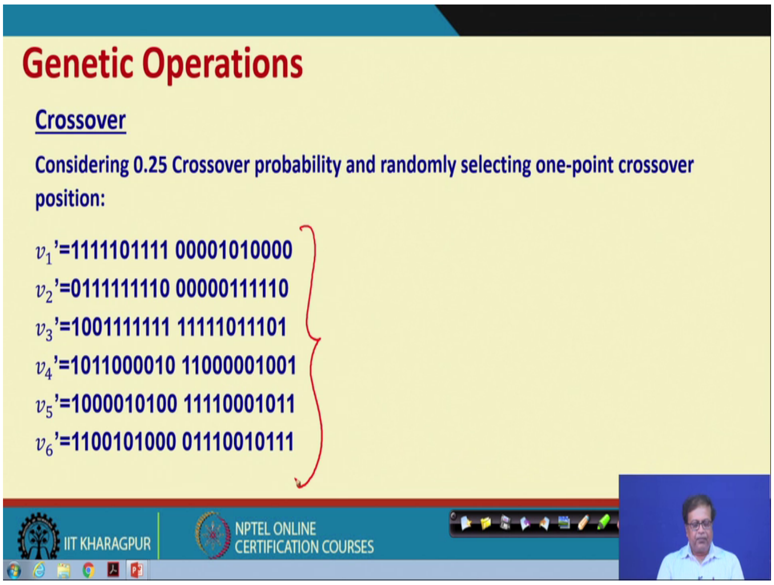
(Refer Slide Time: 06:11)



Now, there are different ways of doing it so, the kind of things that we have done is this, that we know we represent these 6 values in such a manner that first 10 bits represent r. So, this is going to be r and this is going to be h, this is r this is h. Now, you see in the first 10 bits all once means what is it 1024 or 1000, what we have done, we have taken all bits as 1000, we have that is how here represented is it alright. So, because if you if you recall let us look at a kind of solutions that we had taken you see r is exactly 10 right.

So, we have to represent exactly 10s so, all bit 1 we take as 10 and not 10.24 is it alright. So, therefore, all other values will be like that so; that means, we are really taking you know 1000 by 1000 by 1024 as our accuracy is it alright. So, that is our accuracy so, these ratio should be multiplied with so, really speaking this number is really 1024 all ones for r. So, this is 1024 multiplied by these ratio makes it 1000. So, that is how we encode so, we have encoded each of these strings so, 21 bits to each chromosome and that is our parent population. So, this is our parent population.

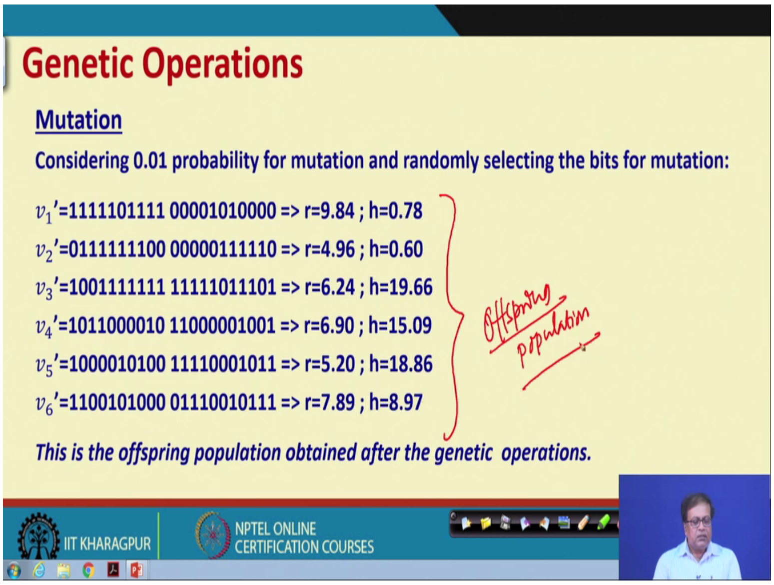
(Refer Slide Time: 08:15)



What next is we should do crossover and mutation. So, arbitrarily through a random number technique with consider 0.25 crossover probability and randomly select 1 point crossover position right. So, if 0.25 crossover probability and we choose some crossover point. So, supposing we choose actually crossover point here right, supposing we choose our crossover point here.

So, you see the first string the first 5 is written raised one is coming from another string. So, you already know how to do cross over we have discussed in detail. So, assume that after crossover we have generated these strings after the crossover operation and, then also we do we plan to do what is known as the mutation.

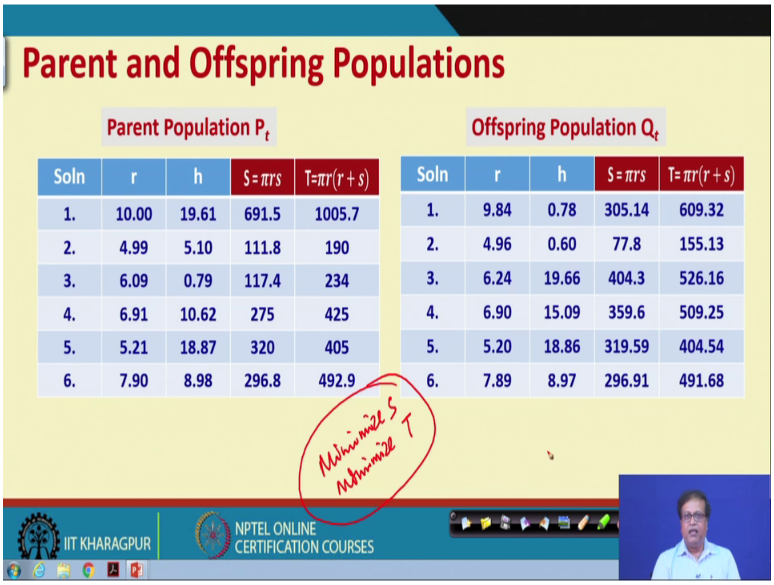
(Refer Slide Time: 09:08)



For mutation what we do we have a 0.01 probability for mutation and randomly selecting the bits for mutation. So, let us say for these numbers you know we have taken a certain values and those values so, this is this is the representation. So, you know we do some mutation after mutation certain 0s and ones are changed based on the probability that we obtain.

And then we get a set of 6 you know feasible solutions, which we can then call as our offspring population. So, what we have done we have started with a representation that is call encoding, then we have done crossover and, we have done mutation and after doing crossover and mutation we have got the offspring population.

(Refer Slide Time: 10:18)



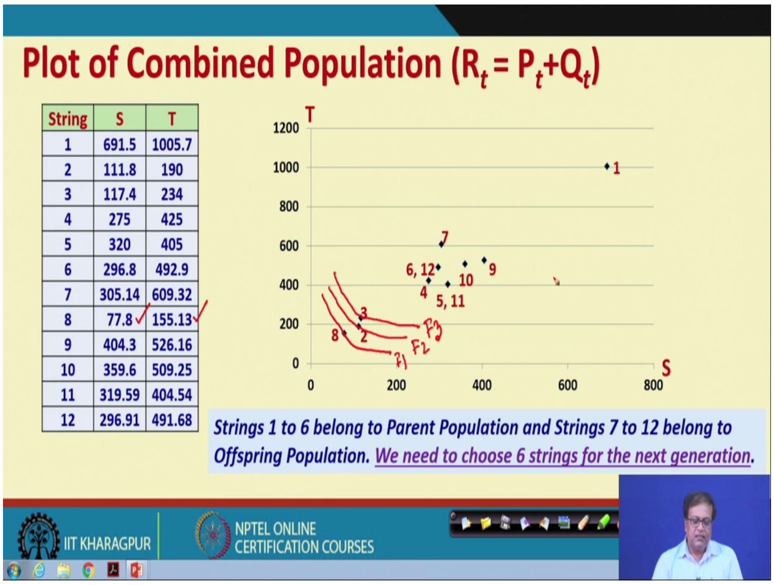
So, essentially we have now got 6 parent population and, we have got 6 offspring population, with different S and T values see minimum our main purpose is that what is our objective function minimize S and minimize T. So, that is what is our objective function so, our objective function is minimize S and minimize T. So, this is our parent population through the crossover and mutation, we have got the offspring population and, we have listed the S and T for the parent and S and T for the offspring with a purpose of minimize S and minimize T.

Now, recall that the total population we have now is 12 strings. So, if we call them as strings or chromosomes and again another set of strings or chromosomes for offspring, a total of 12 strings are with us, how many are we going to have in our next generation 6 of them. So, we have to choose question is how do I going to choose them out of these 12, which are going to go to the next generation, had it been ordinary kind of genetic algorithm the offspring population will become our next generation recall that.

So, parent population is generation 1, offspring population is generation 2. Sometimes a little bit elitism is done, where we do not take all 6 or we do not do so, many crossover and mutation we do may be for 4 and 2 we directly bring from parent population the best ones. But here we do not do that from the 6 parent population, we generate the 6 offspring and out of all the 12, we try to get the non dominated solutions into our next generation.

So, like the example we have shown in our previous lecture. So, out of these 12 we have to choose 6 solutions which are best possible done dominated ranks right. If I get 6 in the non-dominated the Pareto optimal solutions in the first this thing itself; obviously, we take those 6 it is very simple, but if that is not there then what should we do let us see that.

(Refer Slide Time: 13:10)

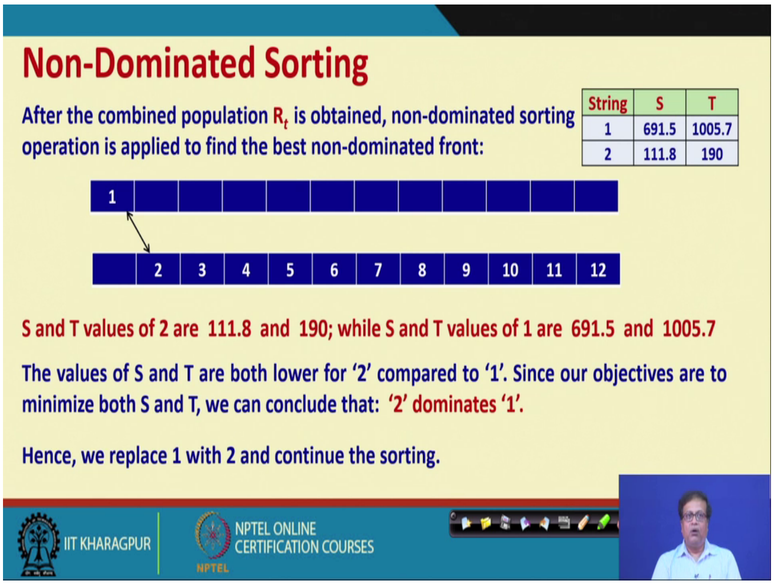


So, this is what we have got supposing we plot so, these are our 12 values, these are our 12 strings and this is; what is our plot. So, if you plot then you see the string 1 to 6 belong to the parent population and 7 to 12 belong to the offspring population, we need to choose 6 strings for the next generation. So, from the plot can you see that 8 is best with regard to S and with regard to T, can you see that you know which 1 is the minimum S that is this one, which one is the minimum T this 1.

So, in the plot also the Pareto optimal front will be like this. So, you can call it as the front 1, now if I take out 8 right, if I take out 8, then which one is the best possible solution which minimizes both S and T right, then again you see that is 3 so, two will be in the front 2, then it will be 3 so, that will be the front three is it alright. So, like this you know we have to go ahead and find the Pareto optimal fronts.

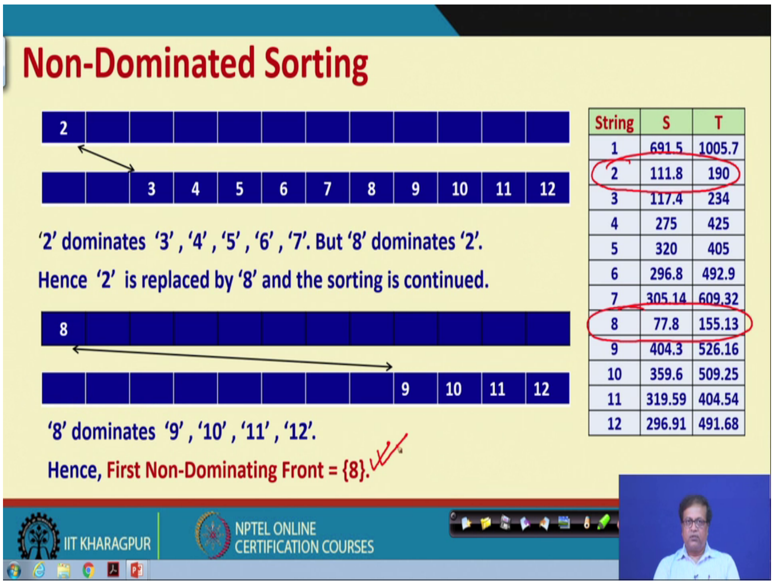
Because had it been only one choice there is no doubt that 8 is the best solution and in the first front only one is there, why only first front one is there, because it is based from both counts is it alright, but only one you will not do we need to have more so, we choose the taking out 8, which one will form the non dominated so, rank 2 that is the front 2, then we will have something what is in the front 3, what is in the front 4 front 5 like this we have to really do that. So, that is how we should choose the 6 strings for the next generation right. So, let us see the procedure the procedure we follow is this.

(Refer Slide Time: 15:30)



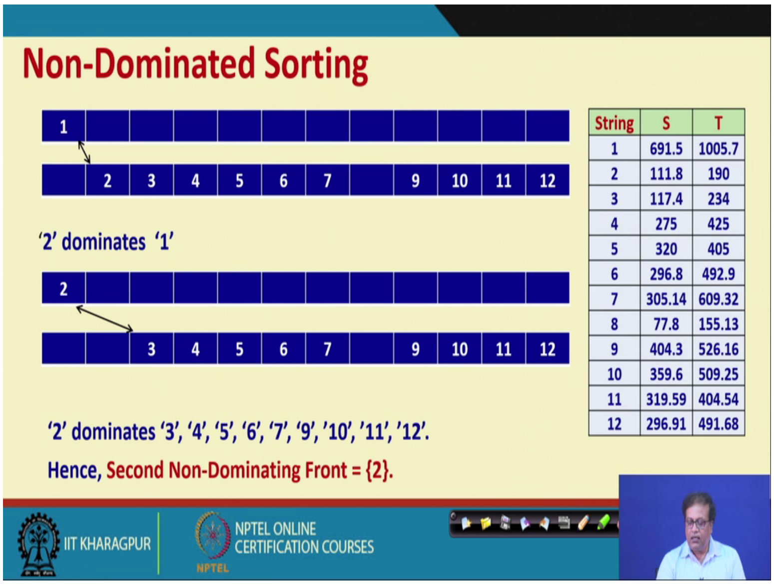
So, we really combine say 1 versus 2 so, if I compare that is how to begin that is the process actually, because everything has to be implemented in computer based solution has to be obtained so, this is how the solution is done. So, if I compare 1 and 2 I see that the values of 2 is better than 1 is not. So, if you compare S and T, then 2 dominates 1 is it alright. So, since 2 dominates 1, 1 will be replaced by 2 alright.

(Refer Slide Time: 16:12)



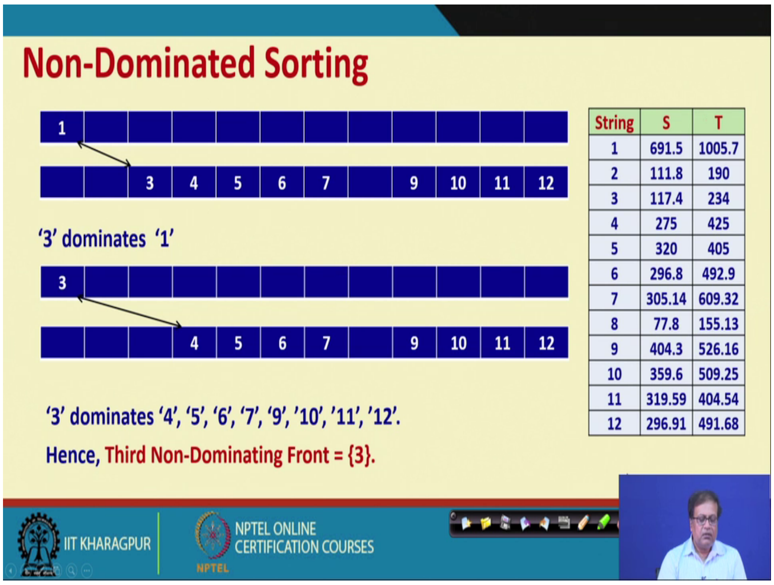
So, that is what is the 2, then similarly if I keep on comparing that 2 dominates 3, 4, 5, 6 and 7, but 8 dominates 2, you know if I compare 2 and 8 so, this is 2 so, this is 2, but this is 8 we see that 8 dominates 2 so, 2 is replaced by 8 and the sorting continuous 8 also dominates 9, 10, 11 and 12. So, therefore, the first non-dominating front is 8 right the first non dominated front will be 8 so, we have got 8 we have seen graphically also that 8 comes in the top so, that is the first that is the front 1.

(Refer Slide Time: 17:02)



Then after that we continue an again we see that one dominates that is 2 dominates 1 and 2 also dominates all the others; obviously, 8 is gone because 8 has been formed of the first front. So, 2 dominates 3, 4, 5, 6, 7, 9, 10, 11, 12 so second dominated non dominating front will be 2 is it alright. So, we have got 2 non dominated fronts that is 1 and 2 right.

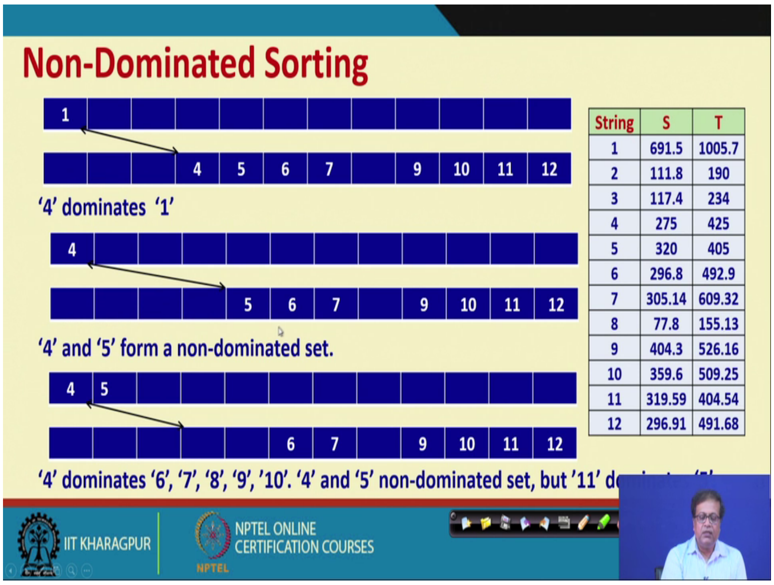
(Refer Slide Time: 17:38)



Now, three if I see 3 dominates 1 and 3 also dominates all the others. So, you know it is coming exactly true you recall what I said that 8 comes first, then 2 then 3 so, 8 is in the front 1, 2 is in the front 2, 3 is in the front 3, you see it is not like in the previous example where we got 4 of them in the first front itself. Here it is not like that 8 is a clear winner and take out taking out a 2 is a clear winner, taking out 2 3 is also a clear winner.

So, that is the front 1, front 2, and front 3 so, we got 3 fronts, but then we need 6 of them for next generation recall that, that we had 12 here 6 first 6 are parent next 6 are offspring out of 12, 6 will go to the next generation. So, even if we take first front second front third front we have got only 3 we need 6 of them.

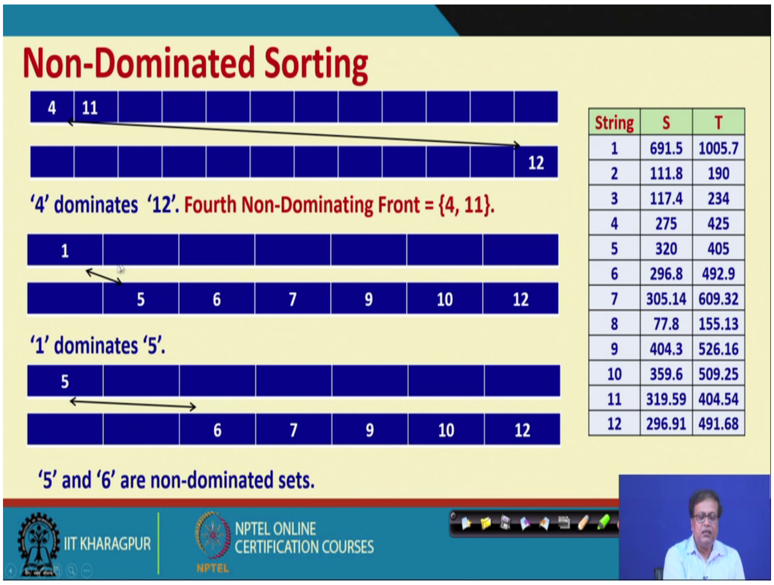
(Refer Slide Time: 18:46)



So, again we compute continue living out those 8 2 and 3. So, you see 4 dominates 1 4 and 5, but if we compare 4 and 5 look here, the 4 and 5 you know 4 is better than 5 in S, but 5 is better than 4 in T right. So, that is what we see that between 4 and 5 there is no winner. So, they form a non dominated set is it alright so, 4 dominates 6, 7, 8, 9 and 8 is not there; obviously so, there is a mistake there is no 8 4 dominate 6, 7, 9, 10, but cannot dominate 11 although 4 is better than 12.

So, you look at 4 and 12 so, 4 is definitely better than 12 so, 4 is better than all others excepting 11 compare 4 and 11. So, where 4 is better than S, but 11 is better than 4 in T. So, 4 dominates 6, 7, 9, 10, but 11 dominates 5 and 4 does not dominate 11.

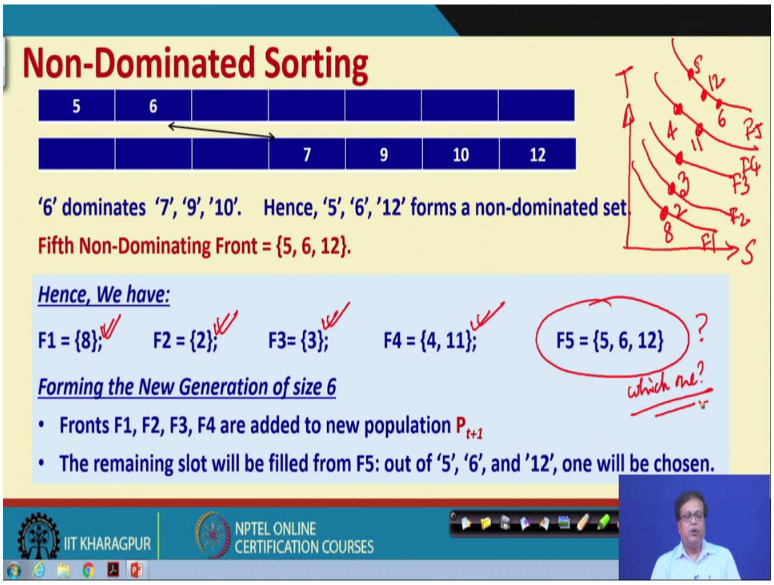
(Refer Slide Time: 20:16)



So, all these facts really point to the fact that, that at the next this thing, the forth non dominating front will be 4 and 11 is it alright. So, because you know 4 is better than all, but 4 in 11 they are forming a non dominated set is it alright. So, that comes to the fourth non dominated front that is 4 and 11 again we see the remaining ones.

So, you see all the rest are taken out so, we have now 1, 5, 6, 7, 9, 10, 12. So, again the same procedure we see that 1 dominates 5 right, 1 is now better than 5 is the opposite the 5 dominates 1 right, that is the 5 is better than 1. So, just opposite 5 dominates 1, then 5 and 6 are non-dominated sets, because of their values.

(Refer Slide Time: 21:23)

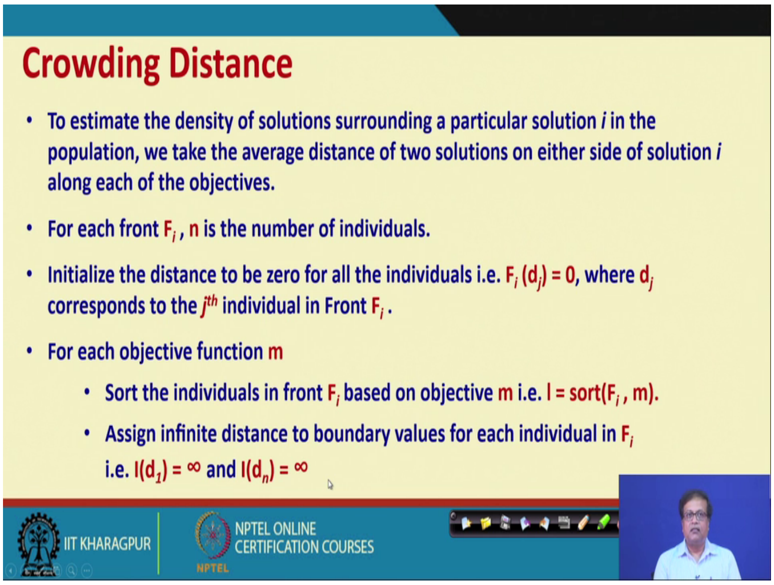


And, then further we see that that 6 dominates 7 9 and 10. So, we finally, find that at this stage 5 6 and 12, they form the non-dominating front is it alright. So, what exactly it means, it means that we have been able to find out this that out of these S and T, we have this one and the first rank we have got 8.

So, this is in the front 1, in the front 2 we have got 2, in the that is F 2, in the third Pareto front we have got 3, in the fourth one we have got 4 and 11 and in the 5th; obviously, the relative positions may not be exactly what I am drawing, but something of this sort right. So, now how many of them 1, 2, 3, 4, 5, 6, 7, 8 are now candidates how many should be chosen 6.

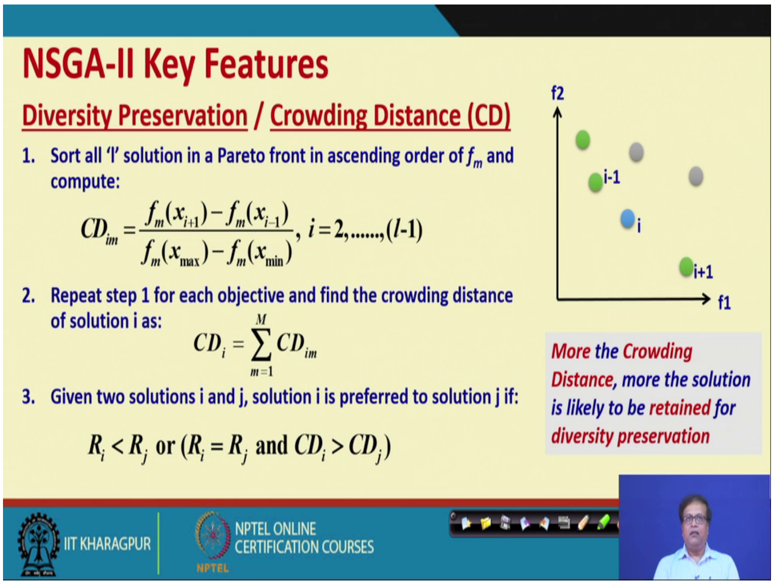
Now obviously, the first thing to choose is rank the rank should be best. So, rank the 8 has got the first priority, 2 is the second priority, 3 is the third priority, then 4 and 11 as the 4th priority so, that makes 5 we need one more question is between 5 6 and 12 which one should we choose that is the question which one is it alright.

(Refer Slide Time: 23:23)



So, they will form the new population that is the question the so, again now recall what we have to do; now we have to check the crowding distance. So, to estimate the density of the solution surrounding a particular solution, we take the average distance of two solutions on either side. So, initialize the distance to be 0, then for each objective m, sort the solutions in front in they that given front and assign infinite distance to boundary values and you know that is the first task, assign infinite distance to the boundary values.

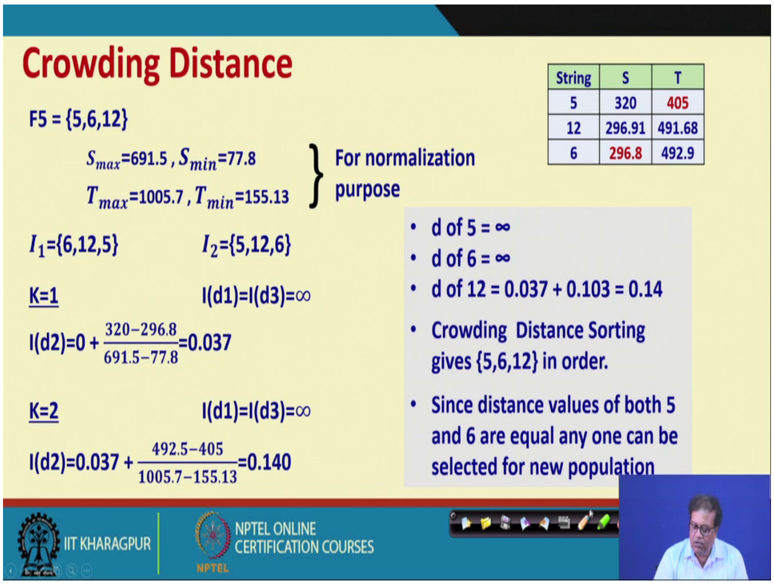
(Refer Slide Time: 23:57)



And for the others this calculation I have already told you, that for each solution find the adjacent values with regard to every objective function. And, then the difference between the objective function value of x i plus 1 minus x i minus 1 divided by the max x max minus x mean for that objective function right.

So, that will be the crowding distance with regard to that particular feasible solution with regard to that objective and, then add for all such objective functions that will be the crowding distance. So, choose first by rank and then by hire crowding distance. Why hire crowding distance? Because take the lonely ones, because the lonely ones if you remove a large portion of the objective function space, we might remain unexplored is it alright so, this is the essential idea right.

(Refer Slide Time: 25:08)



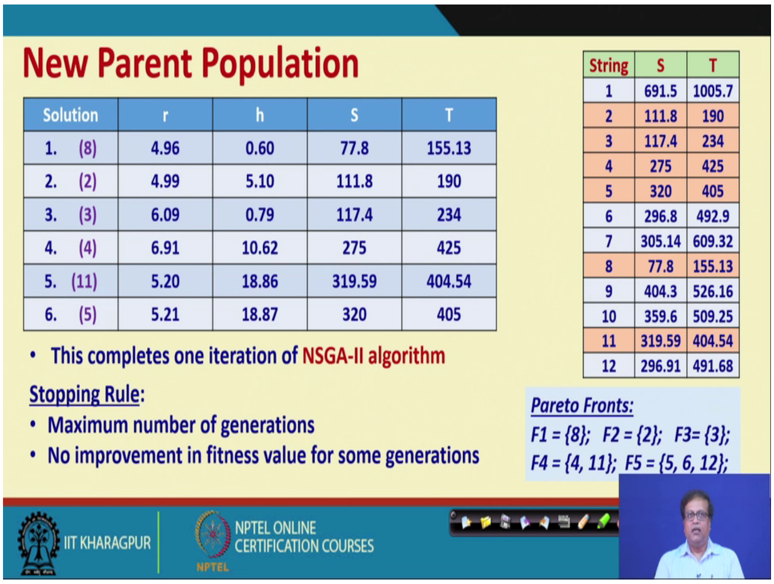
So, that is the conceptual thing, now the actual calculation the question is between 5, 12 and 6, if you if you sort them in S and T, then you see that the best possible T belongs to 5 and best possible S belong to 296.8 belongs to 6. So, you see in a sense with regard to S 6 is boundary and with regard to T 5 is boundary. So, at least with regard to S 5 will be 6 will be given infinite score crowding distance and with regard to T 5 will be given infinite score so, infinity with anything will become infinity.

So, therefore, the crowding distance for 5 is infinity crowding distance for 6 is infinity, what is the crowding distance of 12, see that 12 which is the neighbor of 12, the neighbor of 12 in these particular front where only three elements are there, the neighbor of these are the 6 these are the overall S max and S mean these are the T max and T mean these are the overall values.

So, what is for 12, what is the score for 12? The score for 12 you see with regard to S, this is 12 with regard to 12, the two neighbor values are 320 and 296.8 so, these 2 are taken and what are the S max is 691 and S mean is 77.8, their difference is taken right. So, this ratio is 0.037 and with regard to T see 492.9, 492.5 more or less similar value by 9 minus 405 by these 1005.7 by 155.3 which are T max and T mean.

So, if you take this ratio we get 0.140 alright. So, if you add these and these, then you get the crowding distance of 12 right. So, crowding distance therefore, you see the order is that 5 is the best means highest crowding distance 6 is equal to 5 and 12 is lower right. Since distance values of both 5 and 6 are equal anyone can be selected for new population. So, supposing I choose say 5 alright so, that will now form our new population.

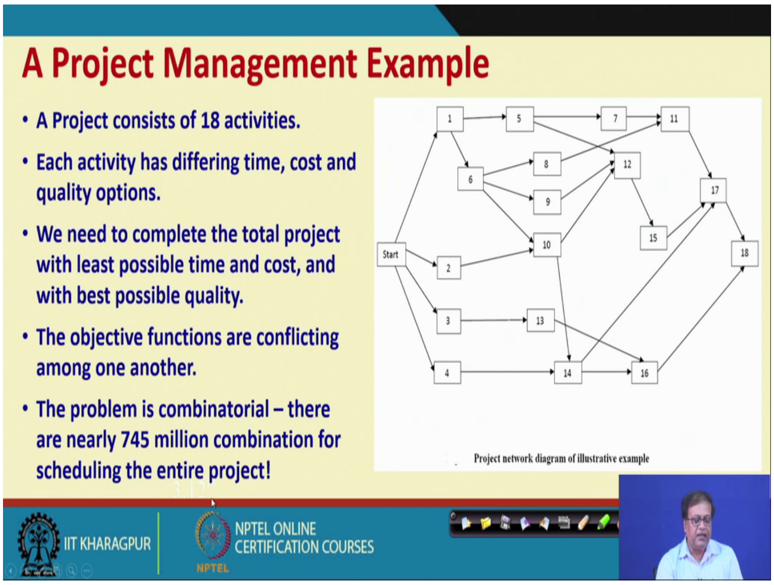
(Refer Slide Time: 28:03)



So, what is our new population? So, look here this is our new population 8, 2, 3, 4 and 11 there in the fourth front and, then 5 is chosen. So, they will form our new population. So, you recall those Pareto fronts 8, 2, 3 so, they are all highlighted 4, 11 and F 5 was 5, 6 12 so, first of all the rank these are having higher ranks, higher in the means lower rank value. So, F 1, F 2, F 3, F 4 they are chosen and F 5 only 1, how it is chosen based on the crowding distance is it alright.

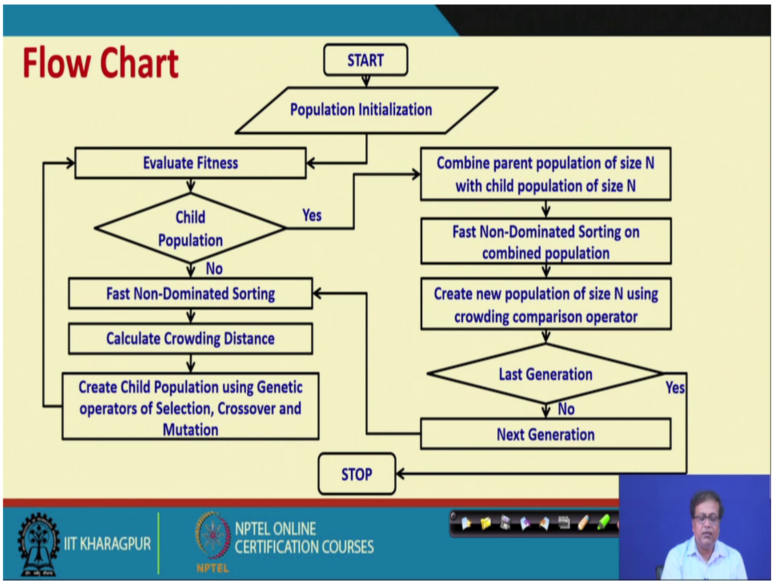
So, when all of them are chosen that completes 1 iteration of NSGA II algorithm is it alright and, stopping rule maximum number of generations, or no improvement in fitness value for some fixed number of generations is it alright. So, this is how we carry out NSGA II.

(Refer Slide Time: 29:13)



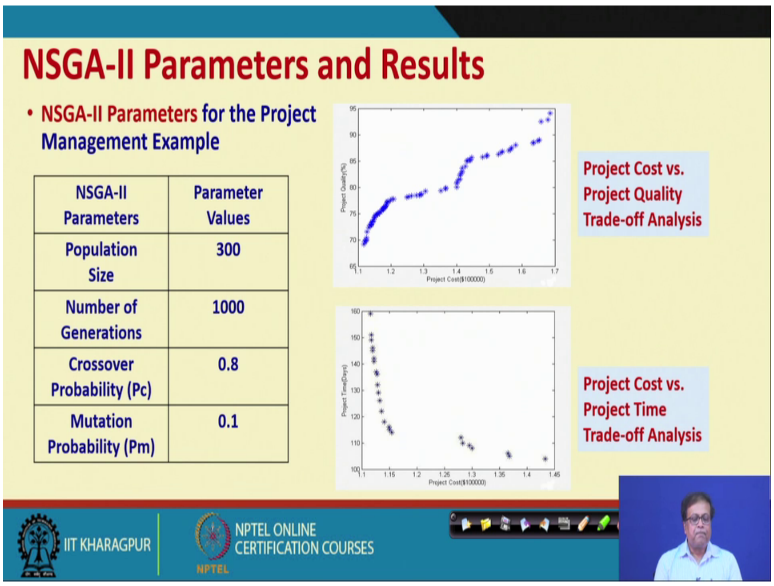
Now, very quickly I will show you as very you know important problem of a project management problem. So, this is a what is called a project management network diagram. So, there are 18 activities and you know we have to if we see really the combinatorial problem, we need to have the time and cost and quality. So, each one has got different options so, combination is very high, you know it could be millions of combinations which are really possible.

(Refer Slide Time: 29:51)



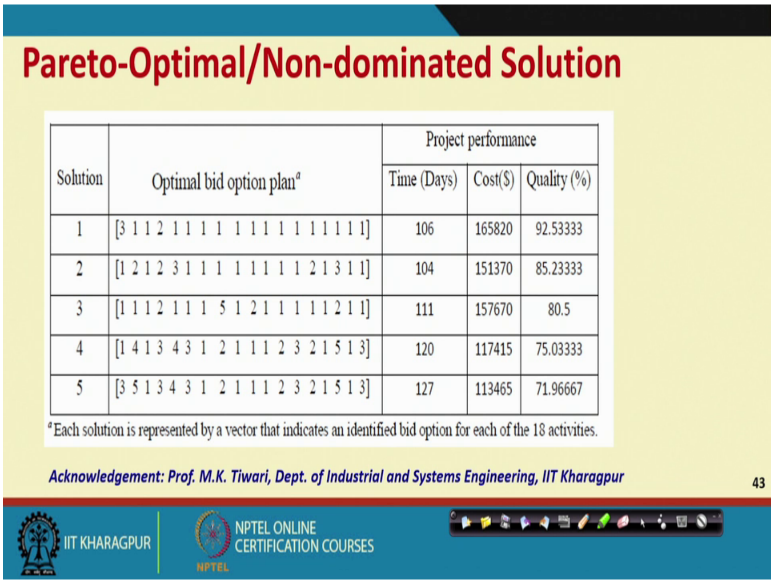
So, this is what we have seen this is how we carry out. So, population initialization evaluate fitness fast non dominated sorting, calculate crowding distance, create child population, is it alright and then keep on going is it alright. Otherwise combine parent population with child population, fast non-dominated sorting of combine create new population, last generation next generation. So, this is how the method already we have discussed exactly how it is done.

(Refer Slide Time: 30:24)



So, this is our NSGA II so, these are the parameters and, this is the project cost versus quality plot that we have got and, cost versus project time trade off that we have got is it alright.

(Refer Slide Time: 30:42)



Solution	Optimal bid option plan[a]	Project performance		
		Time (Days)	Cost($)	Quality (%)
1	[3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1]	106	165820	92.53333
2	[1 2 1 2 3 1 1 1 1 1 1 1 1 2 1 3 1 1]	104	151370	85.23333
3	[1 1 1 2 1 1 1 5 1 2 1 1 1 1 1 2 1 1]	111	157670	80.5
4	[1 4 1 3 4 3 1 2 1 1 1 2 3 2 1 5 1 3]	120	117415	75.03333
5	[3 5 1 3 4 3 1 2 1 1 1 2 3 2 1 5 1 3]	127	113465	71.96667

[a] Each solution is represented by a vector that indicates an identified bid option for each of the 18 activities.

So, these are the kind of results and, if we see the results in the final results, then you see the time, cost, quality, time, cost, quality all of this are that is the kind of best solution that was obtained through the NSGA II. So, you see large combinatorial problems can actually be solved by the NSGA II algorithm is that alright. So, that is where we conclude and, we thank you very much for patiently hearing this lecture right.

Thank you very much.