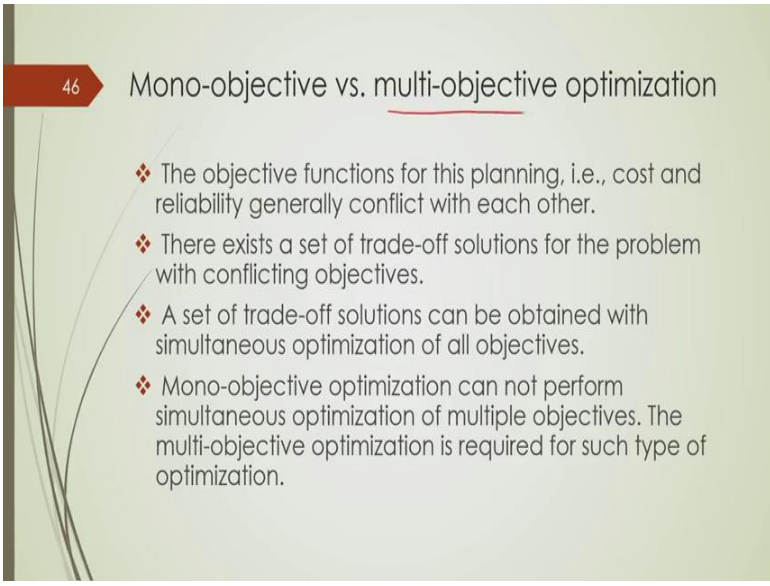
Operation and Planning of Power Distribution Systems
Dr. Sanjib Ganguly
Department of Electronics and Electrical Engineering
Indian Institute of Technology, Guwahati

Lecture - 27
Multi-objective power distribution system planning approach

So, in my last lecture I discuss the single objective or mono-objective optimization problem for solving this power distribution system planning ok.

(Refer Slide Time: 00:46)



And in this lecture, I will talk about this multi-objective optimization problem; multi objective optimization problem. In fact, our objective function etcetera would be same as that of the previous planning problem, but if you can remember my last lecture, you have seen that we had two objective functions formulated.

One is the objective function related to the cost, another is the objective function related to reliability ok. And we formulated two planning problems, two optimization problems. In one problem we solved a planning problem where our objective function is minimization of total cost. In another problem we solved a planning problem, where our objective function was the minimization of total interruption cost. And thereby, we will try to maximize the reliability of the network ok. Now, in general this cost and reliability these two objectives conflict with each other. In fact, I discuss this in the 3rd module, when I was talking about this reliability assessment that this cost and reliability, these are conflicting objectives which conflict with each other. What does it mean?

It means that when we have this cost optimization, we will never get a solution which would be most reliable. Similarly, when we will do optimize this reliability we will not get a solution which will be you know that the best solution or best economic economical solution or best solution in view of the total cost.

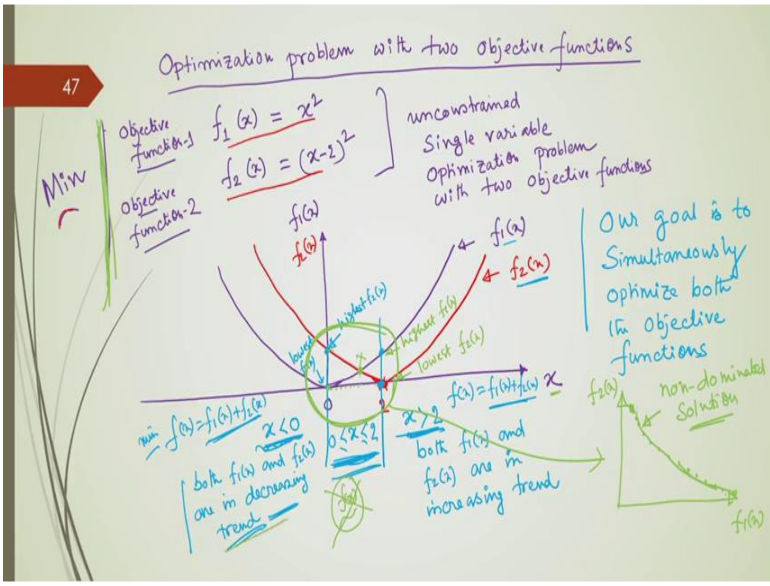
So, there exists conflict-ness. I will talk about this in fact, after a few while ok. In fact, when we have this type of optimization problem, we call this as multi-objective optimization problem, where we have multiple objective functions which conflict with each other. And here, we have the same objective functions as we have formulated in my last lecture, one is total cost another is total interruption cost.

Total cost refers to the total investment and operational cost and total interruption cost is basically the cost associated with all sort of faults or interruptions ok. And when we have this kind of conflicting objectives we use this multi objective optimization approach. It is a very big domain in optimization theory, I will not go into detail of that, but I will give you some insight to understand that what is multi-objective optimization problem. And why we need to have a special care for solving those kind of multi-objective optimization problems ok.

Now, in multi objective optimization problem, unlike the single objective or mono-objective optimization problem, there exists a number of solutions instead of a single optimal solution. And this number of solutions, we are called this trade-off solution or they are known with a special nomenclature that is called non dominated solution, I will come to that ok.

So, in multi-objective optimization, our goal is to have simultaneous optimization of multiple objective functions, which conflict with each other ok.

(Refer Slide Time: 04:51)



Now, I will give an example here. Suppose, I have an optimization problem where I have two objective functions; so, optimization problem with two objective functions ok. This is very simple example to understand the usefulness of multi-objective optimization approach. So, in one optimization, in this optimization problem the one objective is f 1 x, it is a single variable optimization problem where objective function is f 1 x is equal to x square. See this is objective function 1 ok. And another objective function is f 2 x is equal to x minus 5 square let us say ok or x minus 10 square or x minus 2 square or whatever you can call ok. So, let us consider it is as x minus 2 square, instead of x minus 5 square. So, this one is our objective function 2; objective function 2. So, this is a case of unconstrained single variable optimization problem. So, this is un-constrainted single variable optimization problem with two objective functions, with two objective functions ok. And let us consider that our goal is to minimize both the objective functions; our goal is to minimize both the objective function. Now, since this problem is of a single variable, we can easily sketch the function f 1 x and f 2 x with respect to this variable x ok. So, let us do that.

So, this is x starting from 0 to infinity and this in this direction this is negative. Now, this one is suppose, f 1 x. So, if we plot this f 1 x with respect to x, how would be this plot? This plot would be something like this ok. So, this one is our f 1 x ok. Now, I am also plotting here this f 2 x, then how would be this plot?

Suppose this is x is equal to 2. So, this plot of this f 2 x would be something like this. So, this is f 2 x; so this is f 1 x and this is f 2 x. In the same graph, I am plotting both the functions, one is f 1 x another is f 2 x ok. And as you have seen that our goal is here to simultaneously optimize both the objective functions, simultaneously optimize both the objective function ok. So, here our goal is to simultaneously optimize both the objective functions ok.

Now, I will create three regions for these values of your x, one is this region, one is this, another is this. So, here this is basically when x is lower than 0 and this is the zone where x is greater than 2. And this is of course, this is when x varies between 0 to 2 ok. Now, looking at this characteristic, or looking at this graph, one can understand that at this region when x is equal to 0, both f 1 x and f 2 x are in decreasing trend if we increase this value of x ok.

So, if x increases and both variations are in decreasing trend ok. So, here when x is equal to 2, both f 1 x and f 2 x are in increasing trend ok. So, you can see from beyond this x is equal to 2, both are increasing ok. Now, what will happen when x is equal to x is in between 0 and 2? Ok.

If you look at in between this f 1 x; f 1 x is in increasing with the increase of x and f 2 x it is decreasing with increase in x ok. So, in this region, now if we consider that if we optimize this both the objectives at this region when x is equal to x lower than 0. Since both the objective functions are in decreasing trend or with the same type, they follow the similar characteristics we can easily find out that what would be the solution, if we optimize both the objectives together.

And that will eventually similar to an objective function, that f x is equal to f 1 x plus f 2 x; that means, if you simply aggregate this two objective functions and if we minimize that then whatever solutions you are getting that will be the solution of that particular problem. Similarly, same thing is applicable when x is greater than 2, you simply aggregate f x is equal to f 1 x plus f 2 x.

And if you optimize that function f x then whatever solution you will be getting that will be the optimal solution beyond this x greater than 2. But, in this region, when x is in between 0 and 2, since both the objectives are in opposite trend, one is increasing with respect to this variable, another is decreasing with respect to that variable.

Then, we cannot simply aggregate two objective functions and we can tell that whatever solutions we are getting that is the optimal solution that we cannot. Why we cannot? Because, both are basically in opposite trends and or both will conflict with each other.

For example, here you can see at x is equal to 0, that f 1 x is value is 0, but f 2 x if you consider this region f 2 x is of highest value, f 2 x is of highest value ok. So, although f 1 x is of lowest value, but f 2 x is a highest value ok. So, if we independently analyze these two objective function then it may appear to me that f 1 x is getting optimized, the function 1 is getting optimized, but the function 2 is not.

So, when function 1 is giving you the best solution for its own function evaluation, the other function that f 2 x is having the worst solution ok. Now, same thing is applicable when x is equal to 2, that f 2 x is providing the f 2 x that the second function is having the lowest value; whereas, f 1 x is having the highest value ok.

So, here you can see this is highest value of this your f 2 x and lowest value of f 1 x. And here, you can see that the highest value of f 1 x and the lowest value of f 2 x ok. And in between all these points, all these values of x, all these values of x, you can see there are certain points where we have your f 1 x is decreasing, but f 2 x is. In fact, there are certain points you can see that f 1 x is having higher value and there are certain points f 2 x is of higher value ok.

And we cannot simply aggregate these two objectives. If we do so, then we cannot independently find out that which one is having the best solution. In fact, if you aggregate these two then it may so appear to you that this is probably the optimal solution. But, this is optimal solution in view of the aggregation of both the objectives, but not in view of the individual objectives.

If you have the individual goals that I need to optimize both f 1 x and f 2 x then that solution will not provide you the optimal solution either in the objectives ok. And that is why we cannot simply aggregate these two function to have a, we cannot make aggregation of these two objective functions to form a single objectives, we cannot do so because both the objectives behave differently and that is why we call them conflicting in nature. So, if suppose f 1 x is your objective function related to the cost, then you get the best solution at x is equal to 0, but at that if f 2 x is your objective function related to the reliability, then at this particular solution that f 2 x is the worst; so that means, when cost is the in view of this objective function related to the cost, if it provides the best solution in view of the reliability provides you the worst solution ok. And suppose if you do not have any preference prior to this optimization, that I would prefer one particular objective over other, then you cannot decide that which one should be the best solution among these two. And that is what you know goal of having this you know multi objective optimization. In multi-objective optimization, suppose if we have this type of conflicting objectives we do not arrived at a single solution. Rather, we say that each and every point might be one solution for this one perspective solution for this optimization problem. And if we plot these solutions in the objective function domain by keeping this f 1 x and f 2 x in two axis of this plot, then you will get different solutions which will be something like this, which will be something like this. In fact, all you know points of x will have a feasible solution.

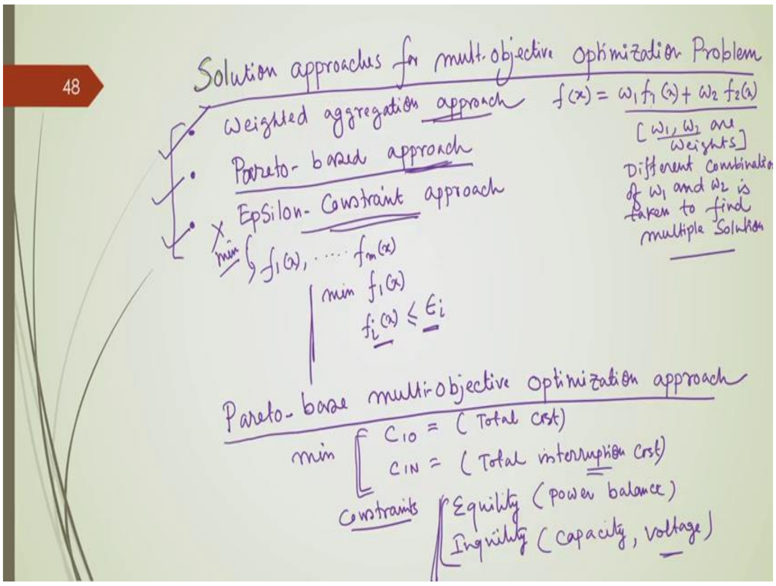
So basically this plot would be a continuous plot like this, where each and every point will represent a prospective solution. So, this type of plot, this type of plot is called in multi-objective optimization, this type of plot is called non-dominated solution. And a set of optimal non dominated solution is called this final solution of a multi-objective optimization approach ok.

Now, so one needs to understand one thing that this will work when we have two objective functions and both the objective functions will conflict with each other. Meaning that in one case when you are getting the best solution in view of the other objective functions you are getting it as a worst solution. So, this is probably the best solution in view of this f 1 x, but worst solution in view of f 2 x, because it is having the highest value and vice versa.

Similarly, these two corner solutions of this will give you the two solutions where you get one objective function’s best value, another objective function’s worst value ok. So, this is what you know the goal of this having multi-objective optimization problem instead of aggregating these objectives together like this.

Because it is not possible to have an aggregated objectives when we have two objective functions which are of different trend, which behaves differently, which conflict with each other ok. And we have different approaches for multi objective optimization, for solving multi objective optimization problem.

(Refer Slide Time: 21:33)



So, let me write solutions or rather solution approaches for multi-objective optimization problem. So, there are different types of solution approaches, proposed time to time for this solving multi objective optimization problem and it is a very well known paradigm of research in optimization theory. And, many people across the globe are working on this ok. So, there are some broad categorizations possible on depending upon the different types of multi-objective optimization approaches. One is called at weighted aggregation method, weighted aggregation approach this is one of the approaches I will talk about this, another approach is called Pareto based approach ok; what is this I will also come. Another is epsilon constraint approach or I should write it in words, epsilon constraint approach ok or so and there are other approaches, as well, for example, lexicographic approaches I am not going into detail. So, there are different approaches for solving this multi objective optimization problem. So, one needs to go through the literature of multi-objective optimization problems and multi objective optimization algorithm or solution strategies for multi objective optimization problem, in order to understand that what are the different types of approaches already reported in the literature for solving this multi objective optimization problem ok.

Now, what do you mean by this weighted aggregation approach? In this approach, although we cannot aggregate this multiple objectives which conflict with each other, but here in weighted aggregation approach it is similar to a single objective optimization approach, where multiple objectives are weighted and aggregated.

So, suppose I have a two objective optimization problem, where our objective function is objective functions are f 1 x and f 2 x and we simultaneously minimize these two objectives. So, here we what we do here, we aggregate both the objectives with some weights. So, where w 1 and w 2 are weights. Now, if you do so then; obviously, you will get a one solution for this particular, by solving this particular problem. Now, here as I told you one solution will definitely not represent that the optimal solution when we have multiple objective functions, which conflict with each other ok.

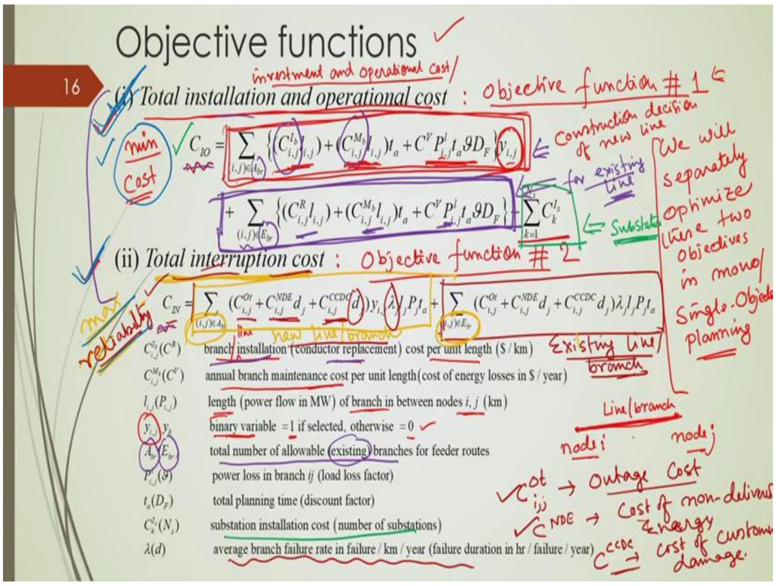
And that is why what they do? They do different combinations of weight in order to get different solution and then plot this. So, this w 1 and w 2 are taken as the values of this w 1 and w 2 are taken as different combination of weights. So, different combination of w 1 and w 2 is taken to find multiple solution, multiple solutions ok.

So, in this approach we need to run this optimization algorithm multiple times in order to have a multiple solution by assigning different combination of these values of the weights ok. Now, this is weighted aggregation approach, but there are some merits demerits which you need to learn by going through the literature of multi objective optimization theory.

Now, there is another approach that is called Pareto based approach, which I have used for solving this multi objective distribution system planning problem, I will come to that. And there is another approach which is called epsilon constraint approach in which even though we have multiple objective functions, we convert. Suppose we have this f 1 x to f m x, we have m number of objective functions for a particular problem in epsilon constraint method, we optimize only one objective function. Let us say this is a minimization problem, we keep only one objective function and the rest of the objective functions are converted to some constraint. So, f 2 x, we convert to some constraint. So, let us write it as f i x to constraint, where this constraint is made with this value of epsilon. So, epsilon i represent the constraint for this ith objective function. And this value of epsilon i is suitably chosen so that we get a multiple solution by varying different values of this epsilon. So, this is what this epsilon constraint approach. We will not be discussing this here, one needs to go through the literature of multi-objective optimization to understand this in a broader way.

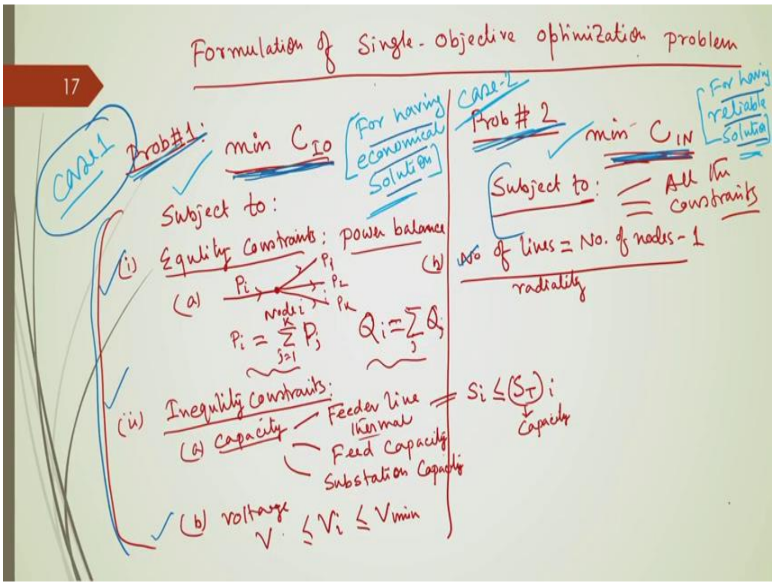
So, here we will be using this Pareto-based approach, in the next slide we will be shown you this Pareto based a multi objective optimization approach. Now, this approach why it is called Pareto based approach? Because this approach is proposed by Professor Pareto, who was a French Economist and Italian born French Economist who proposed this Pareto based approach for multi objective optimization problem. Now, again we go back to this you know this main problem so that I can let you know what is our optimization problem all about.

(Refer Slide Time: 30:17)



So, if you can go back you can see. So, we had two objective functions formulated, one is called this total installation operational cost that is C IO, another is total interruption cost that is C IN.

(Refer Slide Time: 30:31)



And when I solve this problem as a single objective optimization problem, we considered both the objectives at a time and we formulated two of optimization problems, one is problem 1 another is problem 2.

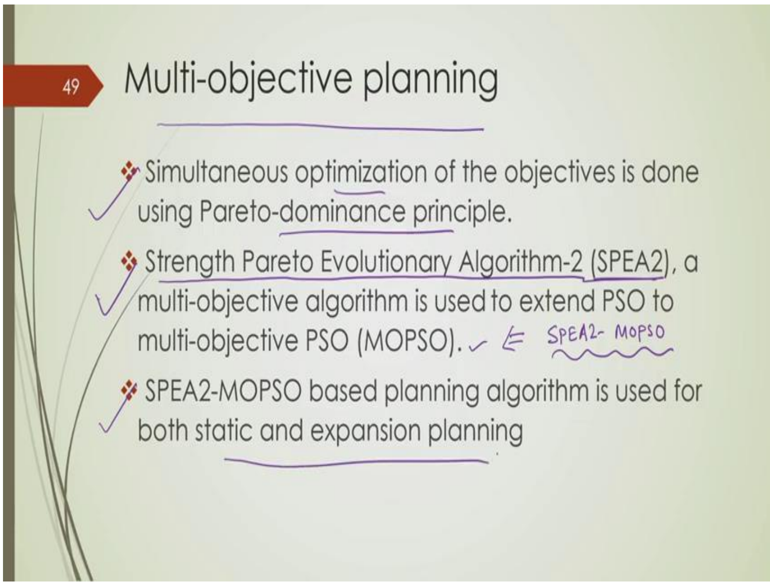
Now, here the advantage of this multi-objective optimization approach is that we will simultaneously consider both the objectives and we can solve the problem as a whole. So, here our goal is to simultaneously optimize both the objective. So, here our goal is to simultaneously minimize C IO and C IN and of course, these two are you know two different objective functions, one is related to total cost which includes that total investment and operational cost.

Another is related to total interruption cost, which is related to reliability by minimizing what we can get a solution which would be a reliable solution ok. And you know both the objectives we simultaneously optimized under the constraints of equality and inequality constraints. We have equality and inequality constraints; equality and inequality constraints, which already I discuss in that lecture where I discuss this problem formulation; equality constraint is to balance the power of all these distribution nodes. So, this is power balance constraint and inequality constraints are the capacity constraints for individual substation, individual feeders, individual ampacity constraint of this feeder branch.

And also this voltage limit constraint; capacity constraint and voltage limit constraint. So, these are the constraints we have in this problem. So, here our goal is very clear that we need to simultaneously optimize these two objectives under these various equality and inequality constraints ok.

So, in previous example I have shown you that we optimize one objective at a time and thereby creating two different problems, but here we are merging these two different problems in a single problem by not aggregating these objective functions, but, by considering the simultaneous optimization of both the objective functions, ok.

(Refer Slide Time: 33:50)

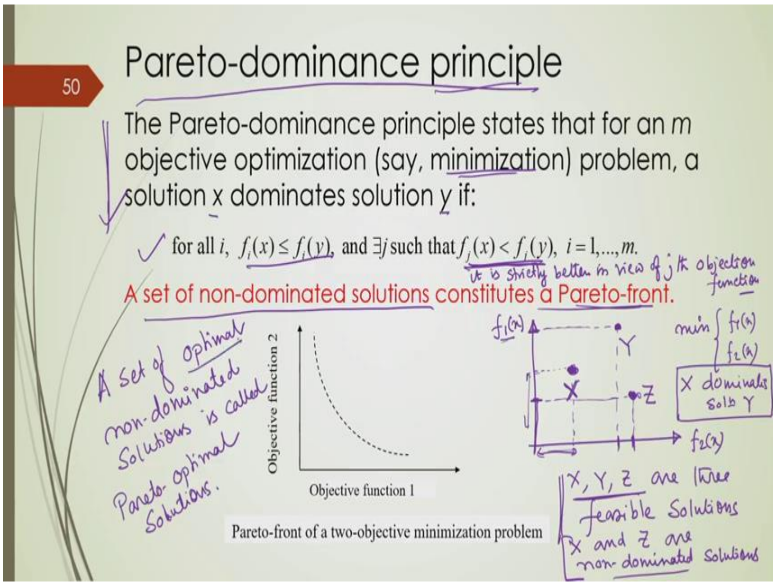


Now, this is what I discuss this multi objective planning is all about simultaneous optimization of objective functions by using this Pareto dominance principle. Again, what is Pareto dominance principle? I am coming to that.

Then we have solved by using a multi-objective optimization approach which is called as Strength Pareto Evolutionary Algorithm-2 (SPEA-2) to solve this multi objective optimization problem for a distribution system planning. And this SPEA-2 is initially a problem which is proposed in genetic algorithm and we have extended this to this particle swarm optimization approach. And so we call this whole approach as Strength Pareto Evolutionary Algorithm 2, SPEA-2 based multi-objective particle swarm optimizations ok. So, we call this open as SPEA-2 MOPSO, which stands for Strength Pareto Evolutionary Algorithm-2 based multi-objective particle swarm optimization ok.

So, we use for both static and expansion planning problem that is for 21-node data as well as 100-node data.

(Refer Slide Time: 35:16)



Now, what is Pareto-dominance principle? So, it is something that one needs to understand very clearly. Now, in Pareto-dominance principle, we call that a solution dominates another solution if it is strictly better, in one in view of one objectives and in view of the other objective it is equally good or it is not worst as compared to the other solution.

Suppose, I have two objective functions, one is f 1 x, another is f 2 x. And I have two solutions, one is capital X, another is capital Y ok and also one is capital Z. So, X, Y, Z are three feasible solution feasible solutions ok. So, now, in this Pareto-dominance principle, according to this you know domination principle, which is mentioned over here, we call a solution x dominates a solution y if it is strictly better. This is, suppose, minimization problem so this condition means it is strictly better in view of jth objective functions objective function ok and it is equally good means it is not worse in view of other objective function as mentioned over this condition.

So, if both the conditions will satisfy then we call the solution x will dominate solution y. So, in order to visualize this let us have three possible solution or three candidate solution or three feasible solutions which represent with this dots ok. Now, as you can see for the solution x this is the value of this you know in view of this objective function 1 and this is the value of this objective function 2.

This is the value of objective function 2 and this is the value of objective function 1. Similarly, for this solution y this is the value of objective function 2 and this is the value of objective function 1 and for this solution Z, this is the value of objective function 1 and this is the value of objective function 2.

Now, you compare these solutions one by one ok. So, you can see this if you compare the solution x and solution y, then you can see since our goal is minimization problem it is a minimization problem and our goal is to simultaneously minimize both f 1 x and f 2 x ok. Now, if so, then in view of this both the objectives you can see X, solution X is providing strictly better solution than solution Y.

So, X is strictly better, if because in view of both the objective functions you can see solution X is having lower values of this objective function ok in view of both the objectives. Now, that is why we call that X, the solution X dominates solution Y.

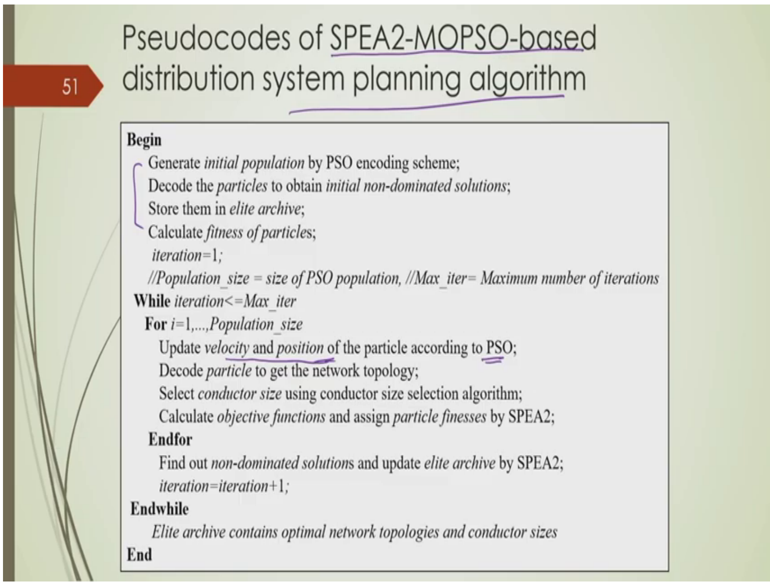
So, we call that X dominates Y, because in view of both the objective function in view of the values of both the objective function X shows the better solution, X shows strictly better solution ok. Because it is having lower value of the solution in view of objective function 1, as well as objective function 2 ok. But, if you compare this solution X with solution Z, then you can see that in view of this you know objective function 1, in view of the objective function 1 because it is a minimization problem.

This solution Z is having better solution, but in view of objective function 2, the solution X is having better solution. So, when we have that type of condition, that one solution is better in view of one objective and other solution is better in view of other objective, then we cannot call that either X dominates Z or Z dominates X.

So, in that case we call X and Z are non-dominated solution; non-dominated are non-dominated solutions ok. So, what we call non-dominated solution? When we have two solutions in which none of them is strictly better than other keeping the other objective function values not worst, so, here you can see X is having this strictly better in view of one objective, but that solution Z is also strictly better than X in view of other objectives.

So, when this type of condition will exist then we call both the solution are not dominated by each other, rather they are non-dominated solution ok. And a set of non dominated solution constitutes a Pareto front and a set of optimal non-dominated solution is called as Pareto optimal solution. A set of optimal non-dominated solutions is called Pareto optimal solutions ok. A set of optimal non dominated solution is called Pareto optimal solution ok, and in multi objective optimization approach our goal is to obtain this Pareto optimal solution that one needs to understand ok.

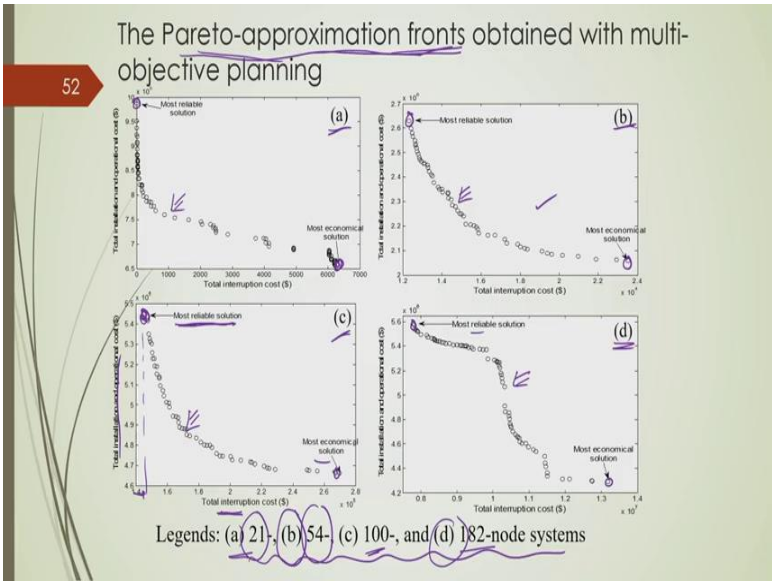
(Refer Slide Time: 42:41)



So, this is the pseudo code of SPEA-2 based multi objective particle swarm optimization for solving this distribution system planning problem.

So, here again, we initialize this population size of this PSO, we also initialize this maximum iteration that should be executed or beyond which this program will be terminated. And we update this velocity and position of this particle according to this PSO principle that already we have mentioned, but here since we have two different objective functions, we have two different fitness functions for this particle and you have seen that we update this velocity and position of the particle in view of the fitness function. Now, here we have two fitness functions ok, and that is why it is multi-objective optimization problem. And by using the principle of strength Pareto, we convert this two fitness functions into a common fitness function ok and then, accordingly we rank the solution; accordingly we find out this non-dominated solution and so and so ok. So, by using this Pareto dominance principle, we find this non-dominated solution and by using this Strength Pareto Evolutionary Approach we assign this fitness function to individual particles and thereby, solving this problem similar to the previous single objective optimization problem.

(Refer Slide Time: 44:21)



Now, these are the results; these are the results that we got for different systems. So, this a represents this that solutions of 21-node system, b is representing this solution for 54-node system, c is representing the solution for 54-node system and d is representing the solution for 182-node system ok.

So, what you can see that this each of this black dot, they represent one candidate solution, one candidate solution and we get a set of solution ok. A set of non-dominated solution, but we cannot call them as a set of optimal non-dominated solution, because that you already I have discussed that in particles swarm optimization we cannot comment on this optimality of the solution.

We cannot say that whatever we are getting at the end that is the optimal solution. So, that is why this set of solution is not Pareto optimal solution, rather they are Pareto approximation solution ok. Pareto approximation solution which might be closer to the Pareto optimal solution and we have done several statistical analysis in order to find this how they are different from one approach to other ok.

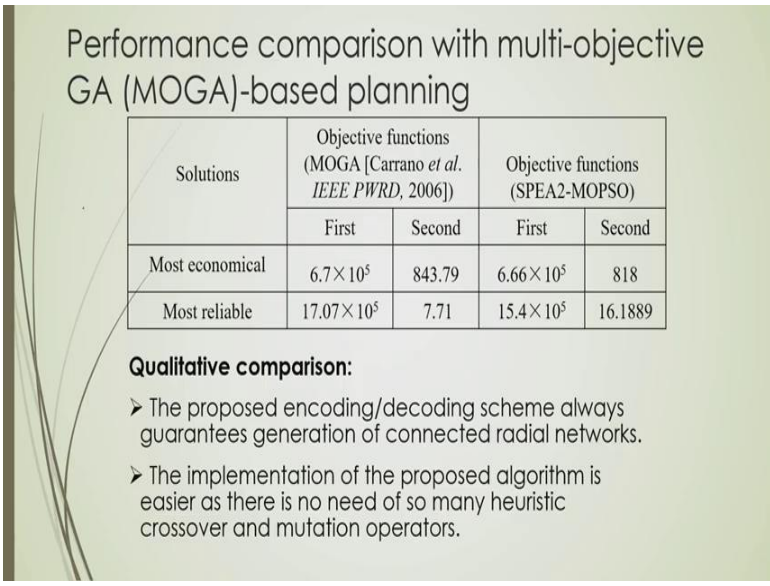
So, another thing you can see that, this nature of this plot of this Pareto approximation solution which is called Pareto approximation front and this Pareto approximation front depends on the different types of problems. So, it is a very much problem specific. So, here it is you know shape is different than this is the shape for the other problem, which is of different planning problem which is of 54 node system.

Similarly, here you get a different shape and here you get a completely different shape. So, the plot of this Pareto approximation solution which is also called Pareto approximation front will show you that how different the different problems are. Also one thing you can see that two corner solutions are marked here. One is this and this for this Pareto front, one is this and this for this Pareto front, and one is this and this for this Pareto front ok.

So, in view of this, you know one objective that is total interruption cost this solution is having you know lowest value, but it is having the highest value in a view of the other objective function. And that is why this solution is the is called as most reliable solution among this Pareto approximation solution, because it gives the best solution in view of this total interruption cost, but worst solution in view of the total investment and operational cost.

So, these two corner solutions are the two best solutions in view of one objective, one objective each ok. So, one solution is the best solution in view of a cost objective, another solution is the best solution in view of reliability objective that one needs to notice.

(Refer Slide Time: 47:57)

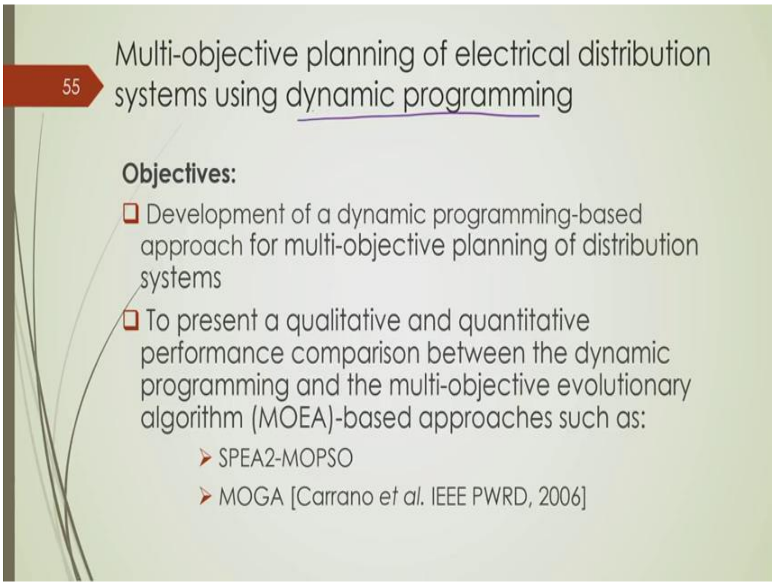


And here also, we have a performance comparison with this paper and this is the source of this our data for this 21-node system and 100-node system. And we got that in fact, we compare these two corner solutions only and we got this for this proposed approach that is SPEA-2 based MOPSO, we have the better solution in view of this economical objective. And in view of this reliability objective, it is slightly higher values for this corner solution. And therefore, we cannot comment that in view of both the solutions; that we are getting the best solution. But, we can say we are getting competitive solutions for this approach ok.

(Refer Slide Time: 48:51)

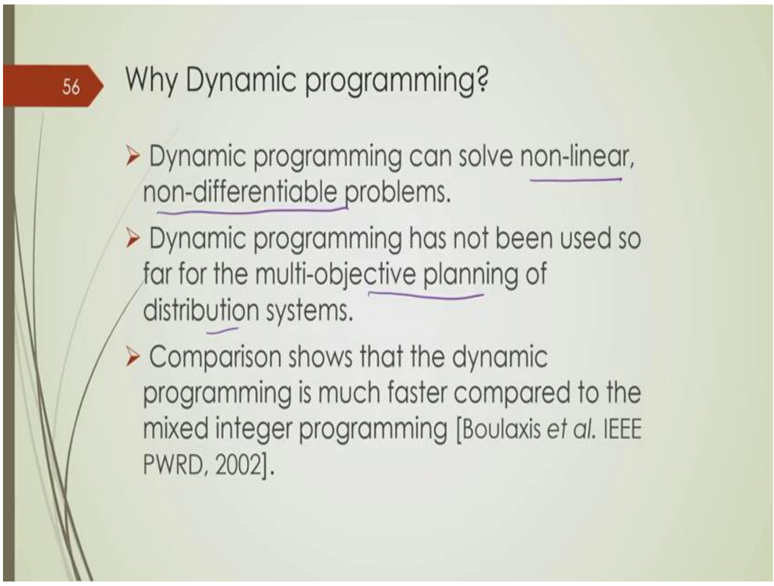
So, let us have a summary over here. So, here up to this, we completed this discussion of mono- and multi-objective distribution system planning problems and corresponding solution strategies, we call them as multi-objective optimization solution. These are discussed for different types of planning problem which includes static problem or expansion problem. Also, different types of encoding, decoding scheme we have used and the performance assessment of several statistical tests, already we have shown in mono-objective problem. In multi objective problem is not shown over here but one can go through the paper where we publish this result, I will show you at the end in the reference list.

(Refer Slide Time: 49:42)



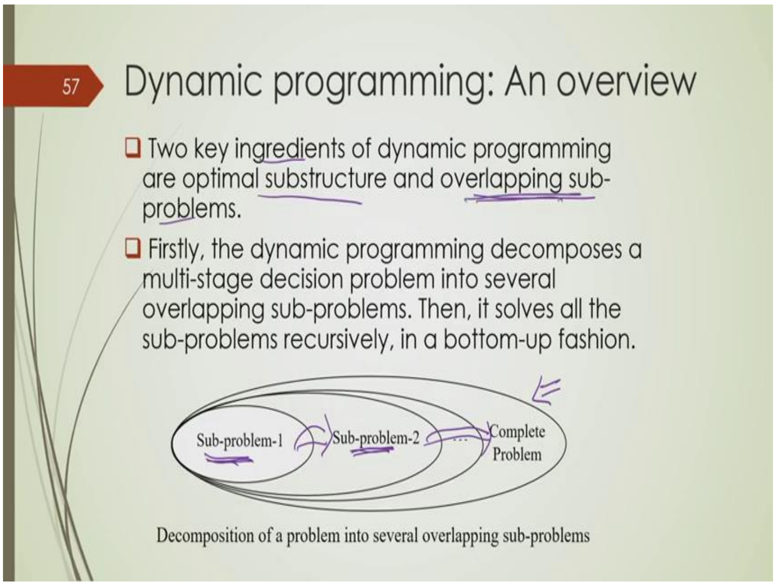
Now, I will talk about another approach for multi-objective optimization for solving this multi-objective optimization problem that is called dynamic planning, that is called dynamic planning or sorry that is called dynamic programming ok. So, dynamic programming or DP, it is a very well known solution strategy; it is a kind of enumerative strategy; it is not a meta heuristic approach which gives different solution after different execution of simulation. But, it provides same solution at the end of this execution and it is a kind of enumerative approach, it works with certain logics ok. And our goal was to have qualitative and competitive performance comparison of this dynamic programming with this multi-objective particle swarm optimization algorithm ok.

(Refer Slide Time: 50:45)



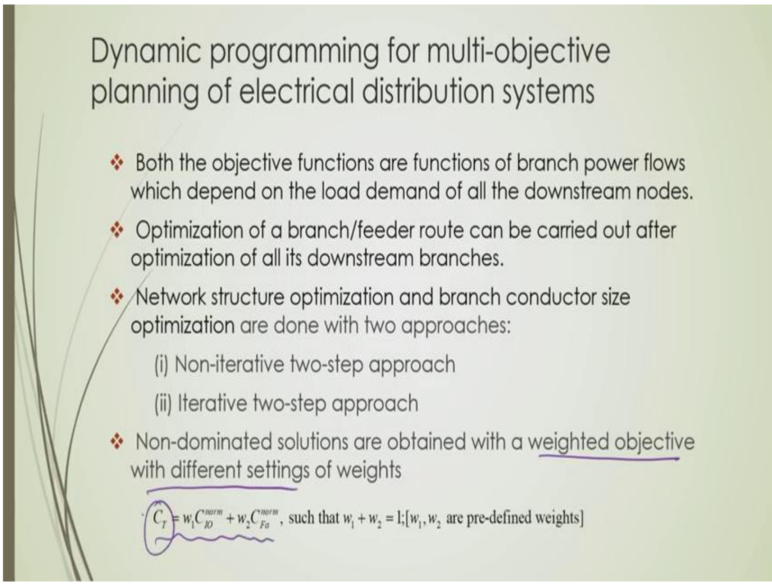
Now, why we have chosen dynamic programming? Because, you know it is not a solution strategy which is very problem specific, unlike linear programming and so. Rather it can work with any type of problem, whether the problem is non-linear or whether the problem is non differentiable. So, linearity does not the requirement for application for this particular approach and we can also extend this approach for solving multi objective optimization, which we have done. And we publish this one also in a reputed journal, I will talk about up in the reference list, also we provide some comparison ok.

(Refer Slide Time: 51:33)



Now, what is dynamic programming? So, basic philosophy of this dynamic programming is: its key ingredients are optimal substructure and overlapping sub-problems ok. Now, what do you mean by overlapping sub-problem? An illustrative example is shown over this figure. So, what it did a complete problem is converted to some overlapping sub problems, like this ok and thereby, this reduces the dimension of the problem or this reduces the overall problem into a set of problems, which we need to solve one by one ok. And this sub-problem 1, you can see, it is independent of sub-problem 2 and thereby you can start this solution of sub-problem 1 first. And this solution will be transferred to this next problem in order to solve this sub-problem 2 and so on. And then finally, we get the complete solution ok. And that is what the philosophy of this overlapping sub-problem ok. So, it starts with a problem which is independent of this all other problems and then one by one we will keep solving the other sub problems as well ok.

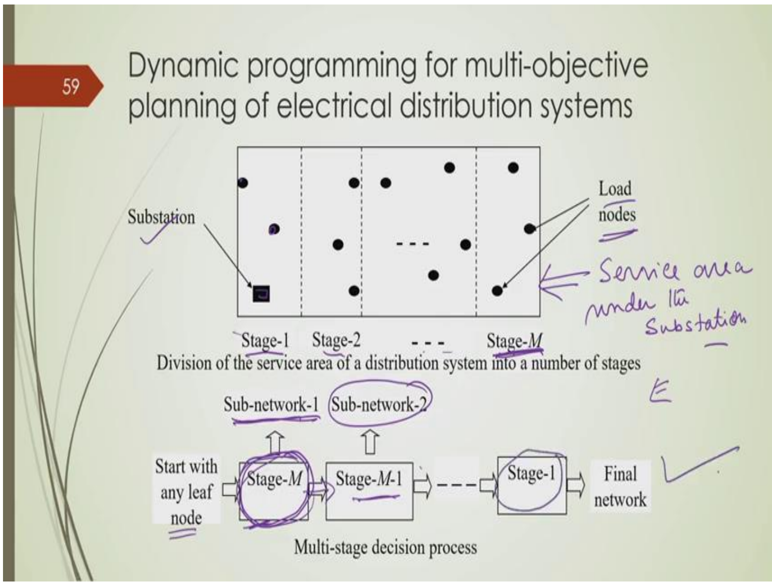
(Refer Slide Time: 53:00)



Now, here this solving about using this dynamic programming for multi-objective optimization problem. Here, we use this weighted approach and we convert these two objective functions into a single objective that is C T and with these two weights, w 1 and w 2, where we use different weight combinations to get a set of solution which we require for to represent in multi objective optimization problem ok. And we use two approaches, one is non iterative approach, another is called iterative approach.

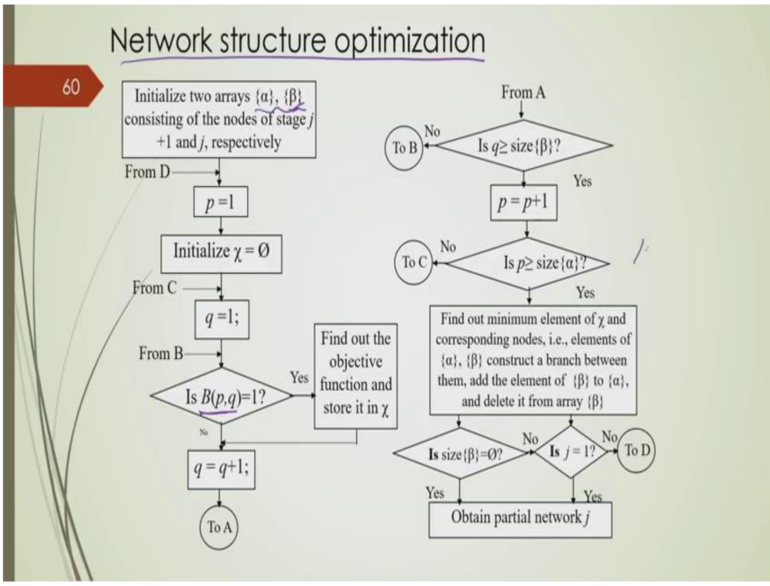
In non-iterative approach this structure or topology of the network is first determined then this branch conductor size is chosen afterwards whereas, in the iterative two-step process both are simultaneously done. So, both the determinations of network structure or network topology and conductor sizes are done simultaneously ok.

(Refer Slide Time: 54:12)



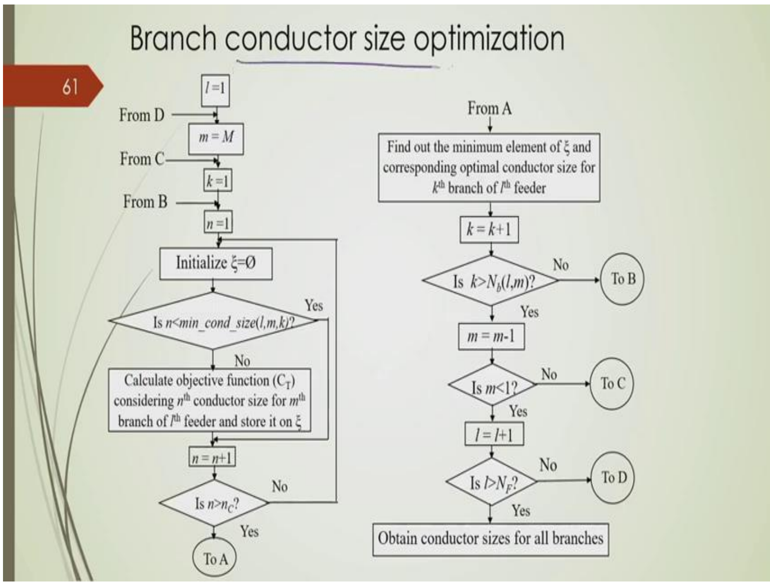
This is an example, how this dynamic programming is used in distribution system planning problem; we divided all this black nodes are representing these loads, loading points or load nodes. And this one as usual is representing the substation and we know the substation location ok. And this is basically service area this rectangle is basically service area under the substation ok. And as per this requirement of dynamic programming we divided this whole service area into number of stages, stage 1, stage 2 to stage M ok. Stage 1, stage 2 to stage M and we will start with a one particular stage as a process of a solution of a sub-problem. So, here we will start with not stage 1, rather with stage M because the philosophy was at the end of this network or the loads or nodes which are located distant away from the substation. They are somewhat independent in the flow of power. So, they only carry those lines which are connecting this distant nodes, they only carry the load currents of individual nodes only which you have seen. But, in substation near to the substation the distribution lines are carrying the loads of all the nodes of a network. And that is what the philosophy behind consideration of the furthest nodes as the first stage to initiate this dynamic programming. So, we will start with stage M and we will solve this sub-problem first ok, we will start with this leaf nodes or the end nodes or the nodes which are the end nodes ok and then after getting the solution we provide it to the stage M plus 1 and this solution is called as sub-network 1 here. So, whatever solution we get by solving this stage M that is sub-network 1 which is submitted to the next stage in order to get sub-network 2 and so on. And then after solving this stage 1 we will get the final network topology.

(Refer Slide Time: 56:54)



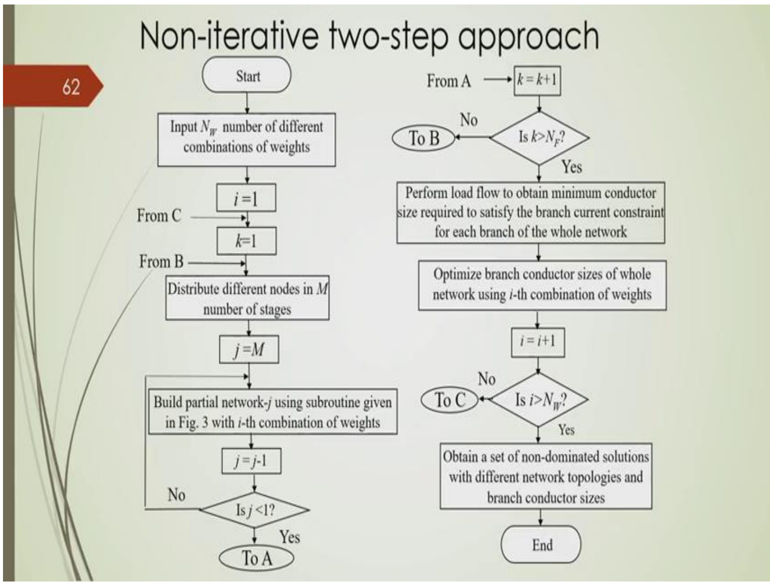
This is the flowchart of network structure or network topology optimization in order to find out the routes of individual lines or feeder branches. Here you can see in order to understand this flowchart, one needs to go through this paper which is published in Elsevier journal, that is IJEPS International Journals of Energy Power and Energy System. So, you can see that we have initialize two arrays, one is alpha and beta and this B is a matrix which is a p defined matrix which assigned that interconnectibility between two nodes. If this is 1, then it is possible to build a network branch or distribution line otherwise 0 represents it is not possible to build a line in between this node p and q. Now, this alpha is initially null for the first stage that is stage M here and beta is consisting of all the rest of the nodes. Now, slowly when we connect one node to the exist this sub network we connect, we put it to alpha and we remove it from this array beta. And that is what is done, if you go through this you will be able to understand.

(Refer Slide Time: 58:24)



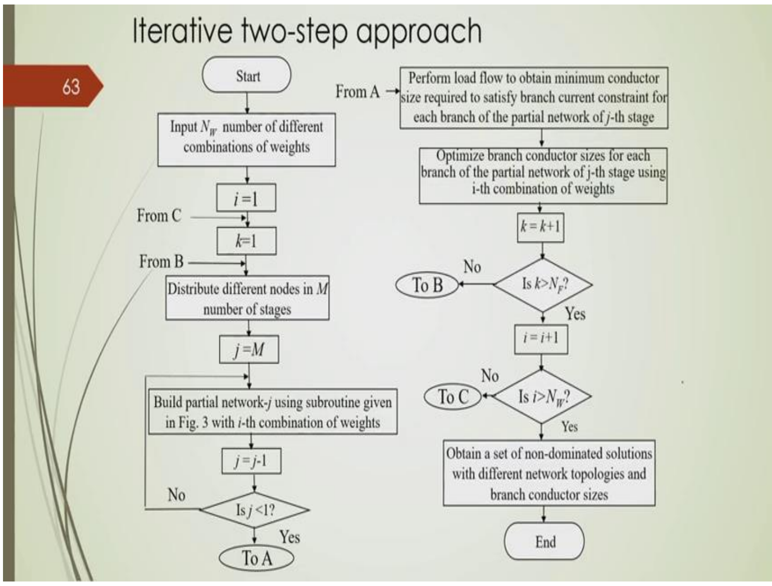
Now, this is how we select this branch conductor size ok, that is the flow chart to select this branch conductor size.

(Refer Slide Time: 58:32)



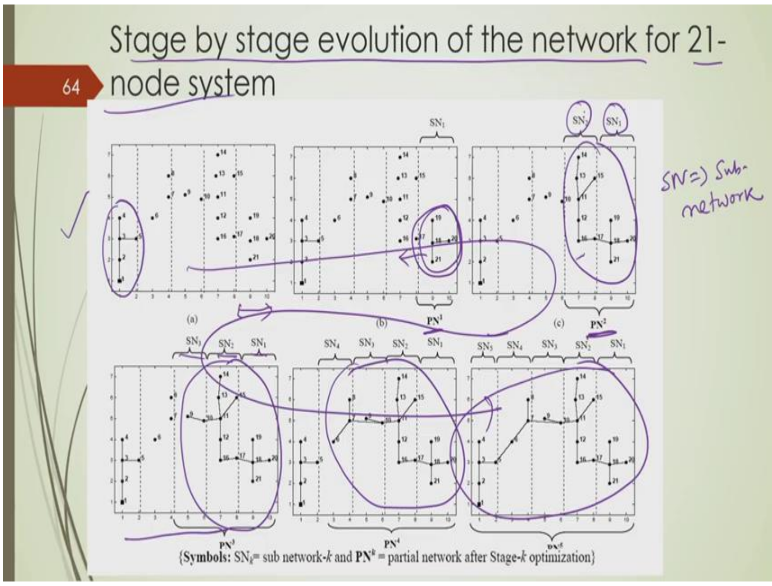
Now, if we do these two sequentially, that we first determine this network structure and then we assign this conductor size, then it is called non-iterative two-step approach. And if we do it together in an iterative approach then it is called an iterative approach.

(Refer Slide Time: 58:54)



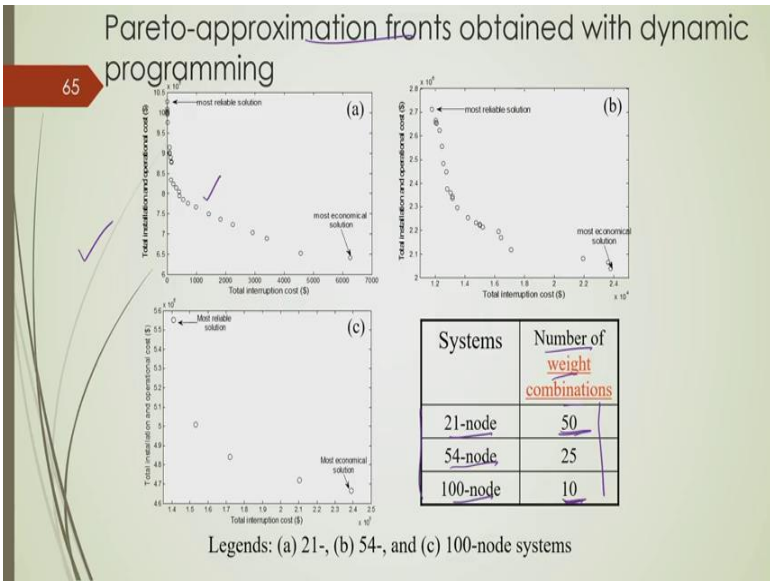
So, you can get a flowchart of both the approaches over here ok.

(Refer Slide Time: 58:56)



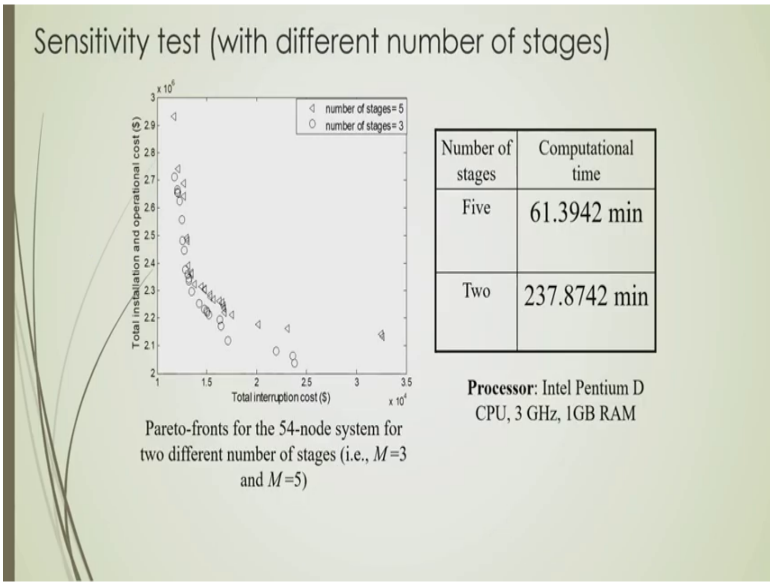
And this is an illustrative example how we get the solution stage by stage for 21-node system, here this is our 21-node system and this is what pre-existing network as you know. So, this is the stage you know the stage where we start this problem. Now, after solving this problem we get a partial node here which is submitted to the next stage. And then we developed another partial network which are represented by S 1 and SN 1, SN 2 and so on. S 1 SN basically represent sub network. So, we first determine this SN 1 in this stage of optimization, then we get both SN 1 and SN 2 in this stage of optimization that is the next stage. And this is submitted to the next stage and we get this sub-network which consists of SN 1, SN 2, SN 3 and then finally, we get this sub-network and then finally, get with this full network ok. So, this is how you can see this network is building. So, network is building up ok by using the philosophy of dynamic programming ok.

(Refer Slide Time: 60:17)



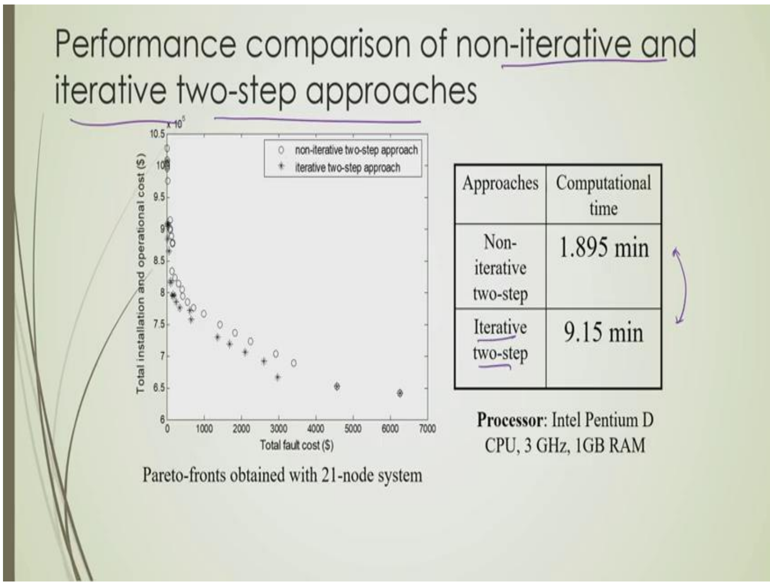
And by varying this weights into different combinations, we get different solutions and we here are the Pareto approximation fronts or Pareto fronts that we get. That is the final non-dominated solution that we got in this dynamic programming approach. So, these are the solutions and these are the different number of weight combinations that we took in 21-node, 54-node, 100-node. And therefore, you can see in 21-node, we get more number of solutions, because we have taken more number of weight combinations. But, for 100-node network, we get less number of solution because we took less number of weight combinations ok.

(Refer Slide Time: 61:07)



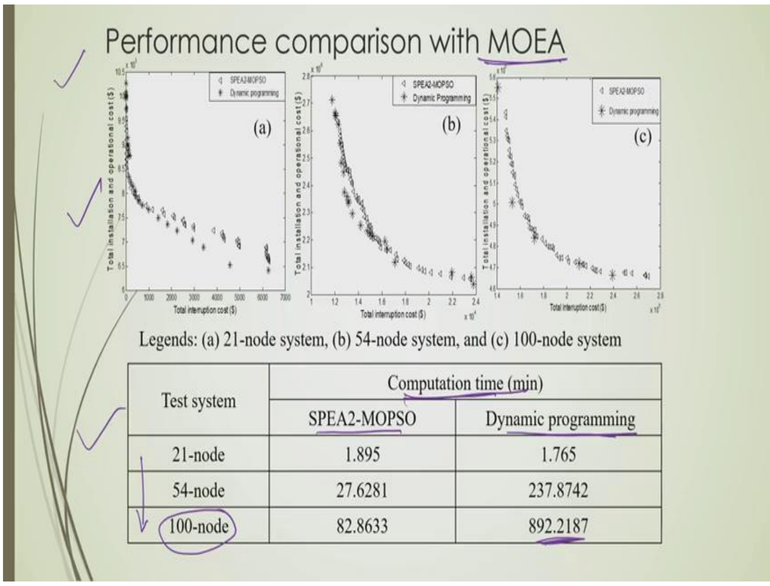
And this is an example for this by varying these stages, this is the sensitivity analysis with varying stages. So, if we have more number of stages we get the computational time somewhat lower and if we take the less number of stages the computational time will be much higher ok.

(Refer Slide Time: 61:30)



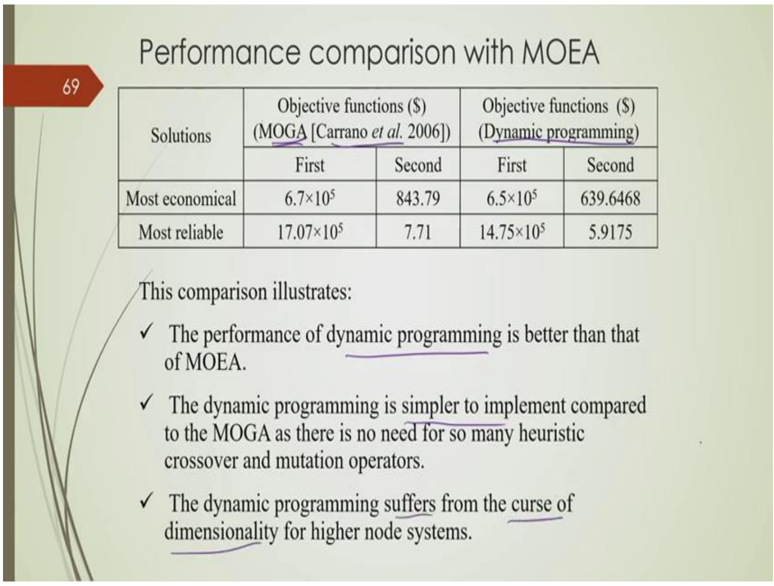
Similarly, this is the comparison of these two approaches, one is non-iterative approach another is iterative two-step approach. There is the comparison of you know computational time, iterative approach of course, takes higher number of simulation time ok.

(Refer Slide Time: 61:54)



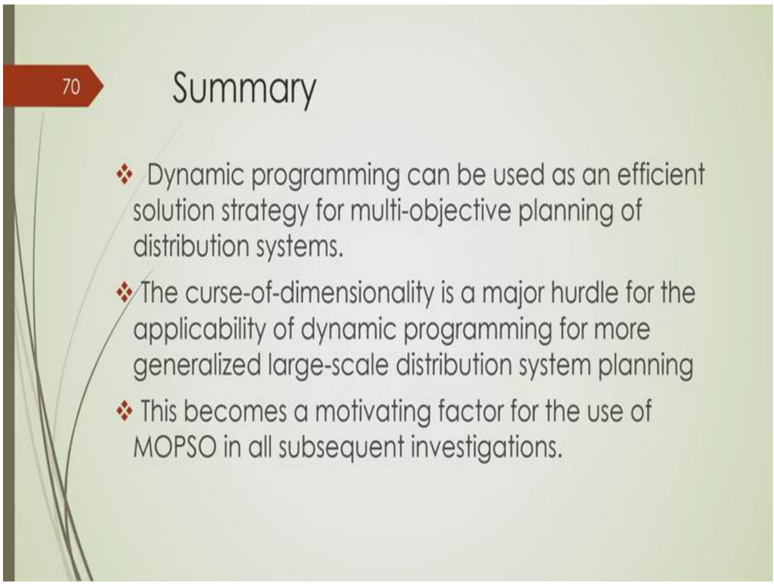
This is the comparison of with multi-objective evolutionary algorithm. So, here we propose this dynamic programming as well as SPEA2 MOPSO and we made this comparison in view of Pareto approximation point, as well as the execution type of computational time ok, which shows that this Pareto front that we got in dynamic programming is somewhat better in different systems, but computational time that the dynamic program takes to execute is much higher specifically, when this number of nodes of the system is getting higher ok. So, for 100-node system, it takes 892 minutes to solve which is a huge amount ok.

(Refer Slide Time: 62:48)



So, based upon that, we can conclude that dynamic programming performs well ok. This is the comparison of this multi-objective genetic algorithm, which I already have shown you proposed by this author in 2006 in IEEE paper and the result that we got for dynamic programming. So, what we get that performance of dynamic programming is even better, and it is also simpler to implement. But, it suffers from the curse-of-dimensionality means that, when we have higher number of dimensions in an problem, it takes a very long time to execute ok.

(Refer Slide Time: 63:34)



So, in summary this dynamic programming can be used as a viable alternative for solving this multi-objective optimization problem, but it suffers from this problem of curse of dimensionality ok. And performance-wise, it provides solutions which are somewhat better than the meta-heuristic approaches ok.

So, with this I will stop today and I will continue in the next lecture.