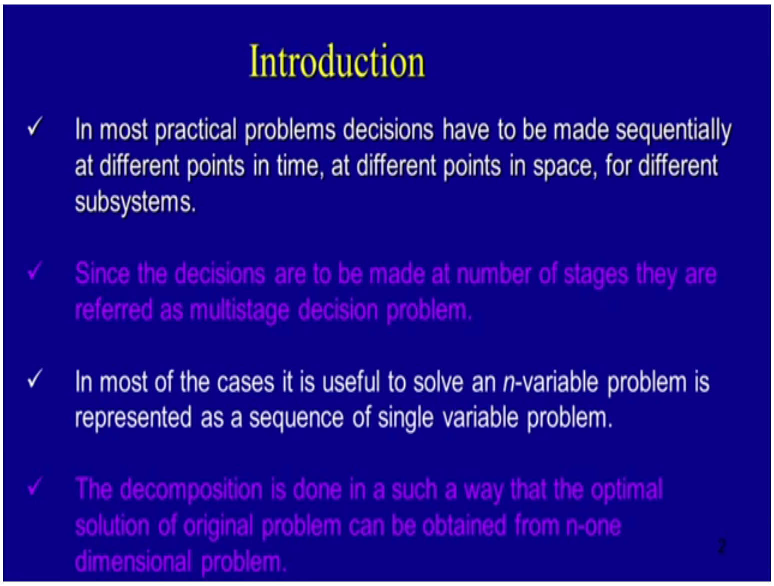
Constrained and Unconstrained Optimization

Prof. Debjani Chakraborty
Department of Mathematics
Indian Institute of Technology, Kharagpur

Lecture – 58
Dynamic Programming

Today the dynamic programming will be discussed, how dynamic programming technique is used for solving the non-linear programming problem, that part I will discuss today. Now in dynamic programming problem there are few properties.

(Refer Slide Time: 00:33)

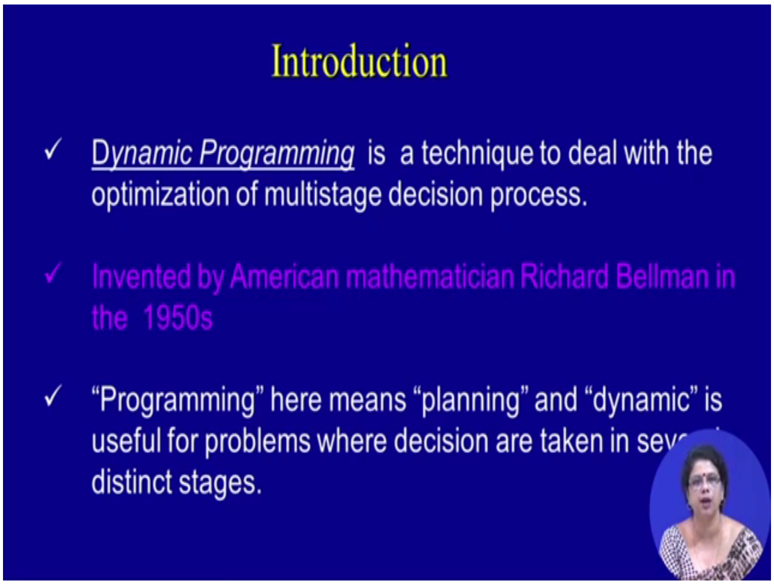


First of all the dynamic programming can be also named as the multistage programming problem because whenever we are dealing with a big problem, it is we have to take decisions sequentially at different points in time, in different points in space and even we can divide the whole system into different sub systems and we take decision for individual subsystem and from there we are finding out the optimal solution for the whole system, but all the decisions are taken in sequence. First we consider the first subsystem; second third in this way we are proceeding, that is why since we are considering different stages that is why this dynamic programming can also be named as the multistage programming technique.

Now, if you have n variable non-linear programming problem, then a set of non-linear programming problem can be solved can be defined as a multistage non-linear programming problem and the multistage non-linear programming can be can be solved sequentially by solving different stages separately.

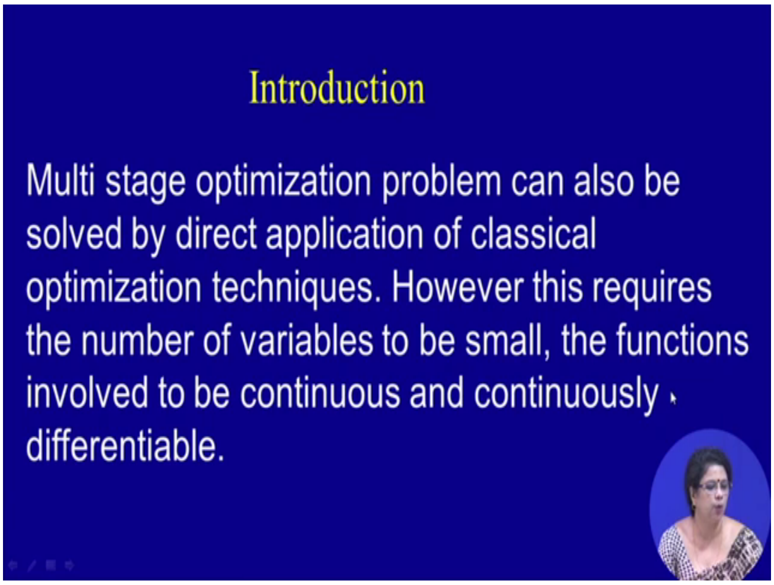
Now, that part today I am going to discuss which kind of non-linear programming problem can be solved using dynamic programming technique, now there are certain basic principles of dynamic programming technique that part I am discussing today. First of all the first part is that, the whole system has to be decomposed into different stages. For decomposition of the system into different sub systems, person should have the skill of doing that then only a n dimensional problem can be defined in 1 dimensional problems, that part you need to understand.

(Refer Slide Time: 02:28)



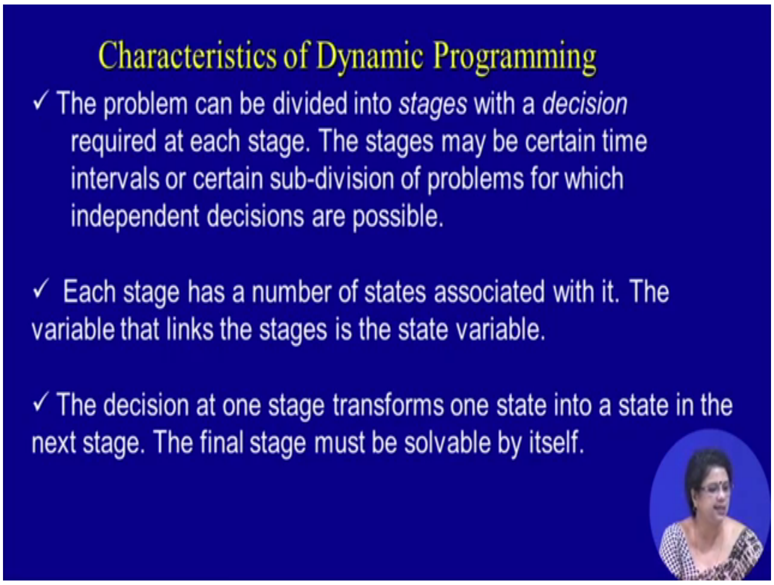
Now, it has been invented by Richard bellman in nineteen fifty and he did dynamic means. So, we are considering the situation in different time scale different space different subsystem different platforms, that is why this is dynamic in nature and programming where using in optimization for planning and that is why this kind of situations, where a problem can be decomposed into different subsystems can be tackled using multistage decision making problem, multistage programming problem or dynamic programming problem.

(Refer Slide Time: 03:15)



Now, the entire dynamic programming problem is being dependent on the will Belmont optimality principle that is why we need to know what the Belmont optimality principle. Now 1 thing is that whenever we are considering a non-linear programming problem, we are considering the objective functions as well as the constraints are continuous in nature and continuously differentiable. If it is not differentiable our classical optimization technique is not applicable there, that is why here also that necessity there for solving the problem for not the non-linear programming problem, we need to have the function as continuous and differentiable.

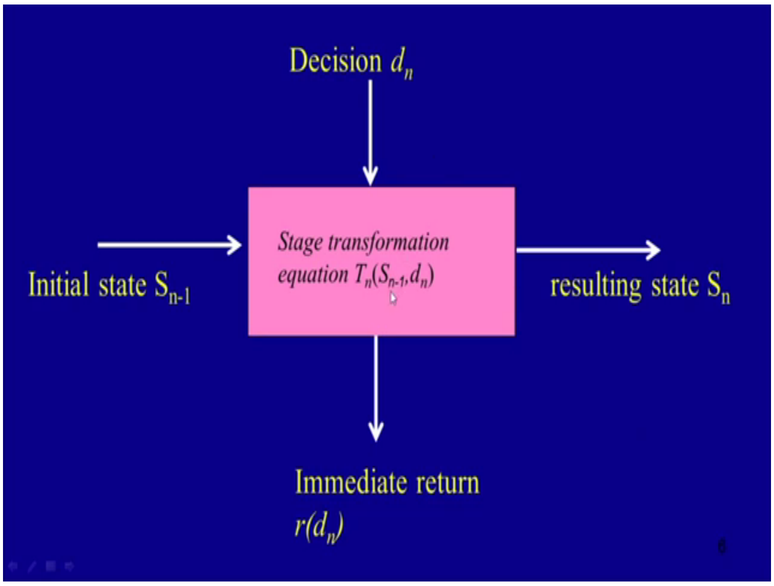
(Refer Slide Time: 04:01)



Now, we will solve in the next, now how to define, how to decompose the problem there a few parts of it. First of all we have to define the problem in to different stages now each stage will have different number of steps and we will have 1 transformation equation in individual stage and this transformation equation will be it we will iterate in different stages to get the optimal solution at different stages and the optimal solutions of the different stages these are connected very much sequentially.

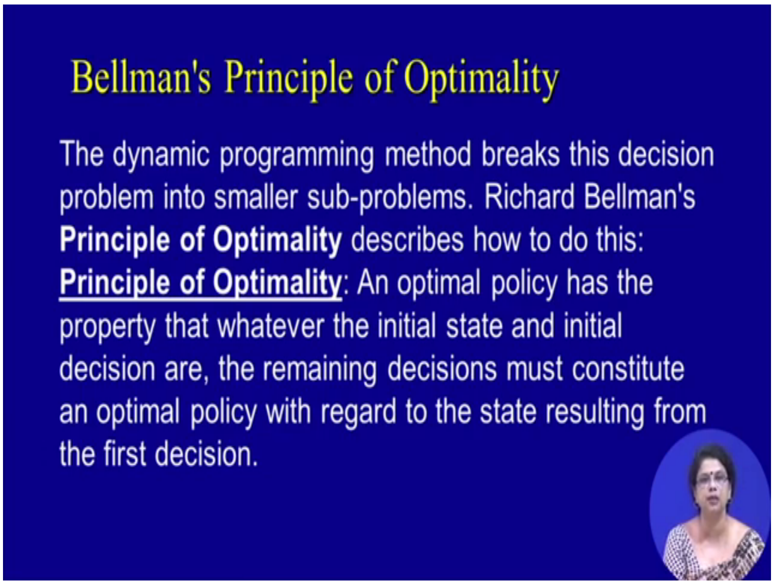
So, that whatever optimal solution we will get finally, at the final stage that will be optimal solution of the original problem. In original problem is the problem, which is big in natural which is very large we are decomposing it in to different subsystems, that is why for every problem our task is to first define the stage then we have to define the states of individual stages and we have to define the transformation equation, transformation strategy in individual stage and how transformation equation is taking role of changing the stages that part I am going to discuss today.

(Refer Slide Time: 05:34)



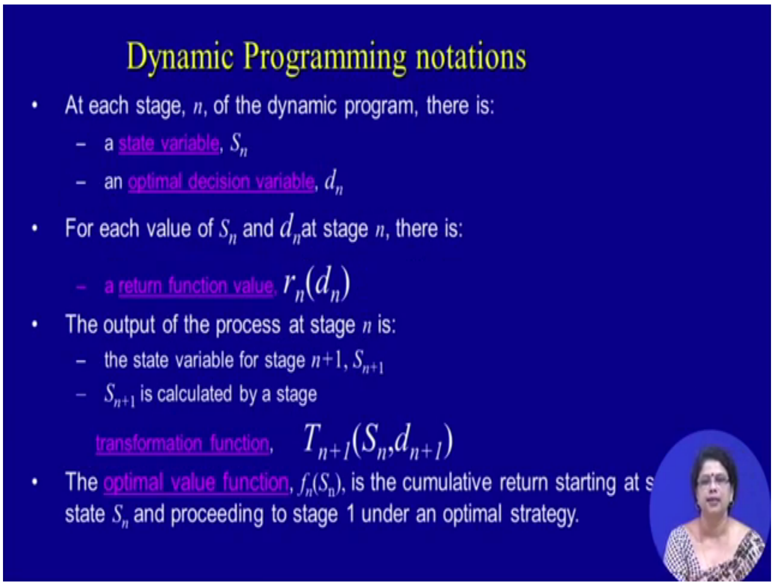
Now, if I just explain it graphically then we will say that each stage that one stage transformation equation will be there, this state transformation equation will be responsible to convert the state S n minus 1 to the resulting state S n and what are the inputs for each, the input will be the decision of the decision of the previous stage and for if previous stage. We will calculate the immediate return and we will use the stage transformation equation to reach to the finals this is the basic idea.

(Refer Slide Time: 06:16)



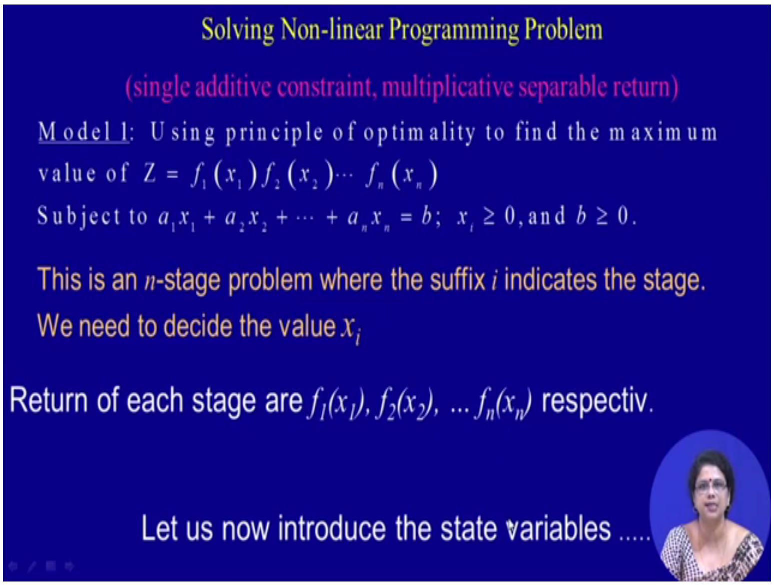
How it is being used I will explain you in the next, but the whole dynamic programming problem is based on the Bellman's principle of optimality. Bellman's principle optimality says that, an optimal policy has the property that whatever the initial state and initial decision are the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision, that is the beauty of the optimality principle of dynamic programming and how this optimality principle is being maintained throughout the whole stage that part I am going to explain.

(Refer Slide Time: 07:02)



Now, there are certain variables certain symbols where using in dynamic programming problem those let me summarize first, now stages are being named as n when n is equal to 1 we are at the first stage, but that is also the convention that the initial state is being numbered as n is equal to 0, now for individual state stage we will have the state variables that is s n and from individual stage we will have the optimal decision that is the d n, this optimal decision will give the return to the next stage. Rather this optimal decision will produce the return, which will be considered in the states transformation equation that is the return is being termed as r n d n, d n if it is the decision variable of the nth stage, then we can see r n d n would be the corresponding return for that decision we can have different values of d n at individual stage, corresponding to d n. We will have different r n d n and by using the transformation equation we will reach to the next stage from n to n plus 1 S n to S n plus 1 all right and the function f is the optimal function will be calculated.

(Refer Slide Time: 08:41)



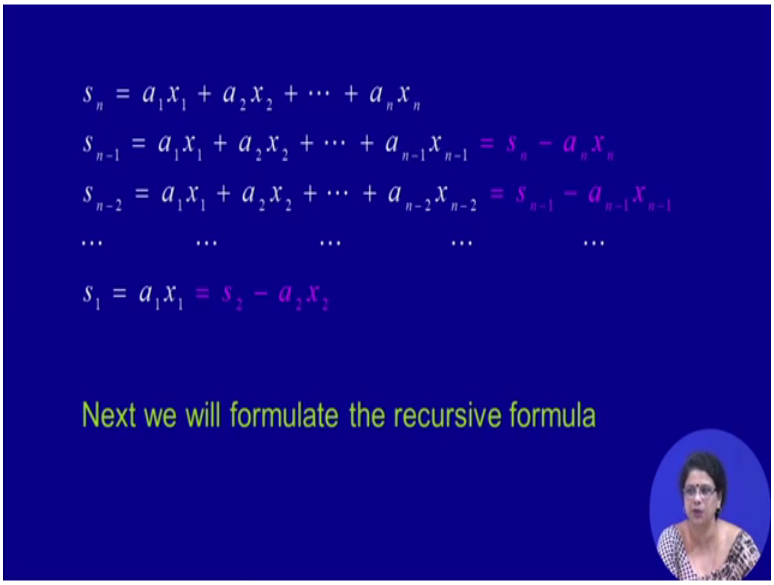
Now, for that thing let me consider one non-linear programming problem, where we are having one additive constraint and we have the objective function that is in the form of multiplicative separable return ok.

This model we are considering for explaining the situation how really we are using the dynamic programming for non-linear programming problem. Now let me consider the model as we have to find out the maximum of f 1 x 1 f 2 x 2 up to f n x n; that means, we are having multiplicative separable radiant return, as I was mentioning before that a separable function is being considered here you see because the objective function can be expressed as different separable functions, which are the functions of individual decision variables. If it is of this form then we can use the dynamic programming to solve this problem.

Now, this kind of problem can be solved with other techniques as well as I have discussed before the constraint optimization techniques, but this dynamic programming very nicely can solve, very effectively can be solved and very easily we will get the optimal solution of it, you see we have consider the objective function as the multiplication of different functions of individual decision variables. Instead of that we can have the objective function as the additive nature, where we will have the functions of individual functions which are additive; that means, we will have the functions in the form of summation in place of product all right. We are considering the simple model here and the constraint is as of the form single additive constant, where having only one constraint this is a simplest model we are considering. Now you see how really we are defining the stage how really we are defining the states for this problem.

Here a 1 a 2 a n these are all the constants b is another constant and Ai s can be positive Ai s can be negative as well same as for b, but x 1 x 2 x n these are the decision variables all right, this is the model for us we are going to apply the dynamic programming technique rather the Bellman's principle of optimality for solving this kind of problem. Now first of all we have to define this problem in to different stages all right, let this problem is considered an n stage problem where suffix I that indicates the stage, that is why we need to decide the value for x i and for each value for x i we are having I is equal to 1 to n we have to introduce the state variables for individual stages.

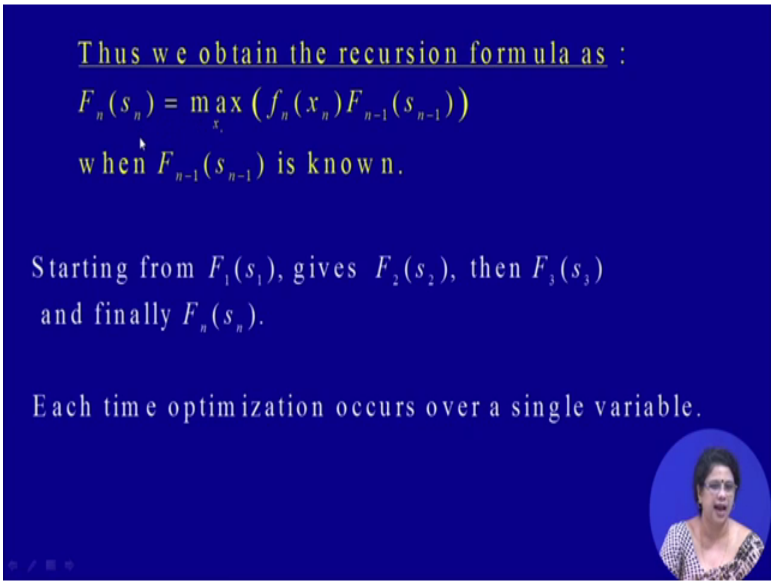
(Refer Slide Time: 11:56)



That is why you see whenever we are considering we are consider considering at the n stage, as n stage as the combination of n variables n minus 1 is stage as combination of n minus first n minus 1 variables. Second stage will be considered that the combination of some additive combination of the first 2 variables a 1 x 1 plus a 2 x 2, thus the first state variable stage variable will be considered S 1 as a 1 x 1. If we considered the state variables like this; we can say s 1 would be is equal to a n a 1 x 1 plus a 2 x 2, that is why this can be written as, S 2 minus a 2 x 2 that is you will get a pattern you see s n minus 1 would be S n minus n by x n minus x n, in this way we will have a pattern to eat and in next we are going to formulate the recursive formula. Now dynamic programming problem as I said, we are having that stage state transformation equation at individual stages and these state transformation equation will play the same calculation, will have the same calculation in each iteration that is why this process is a recursion process and we are going to define next how the recursive formula can be defined from here.

You could see if we define the problem from the constraints state, in this way if we define the variables instead of x 1 x 2 x n, if we just ready define as s 1 s 2 s n we are just transforming the variables from the set of x 1 x 2 x into s 1 s 2 s n. You see we are getting a kind of recursion that there is a recursion process always s n minus 1 would be s n minus x n a n x n, that is why I put x n is equal to 1 e will get 1 set n equal to 2 e will get another set in this way you will get the this way.

(Refer Slide Time: 14:09)



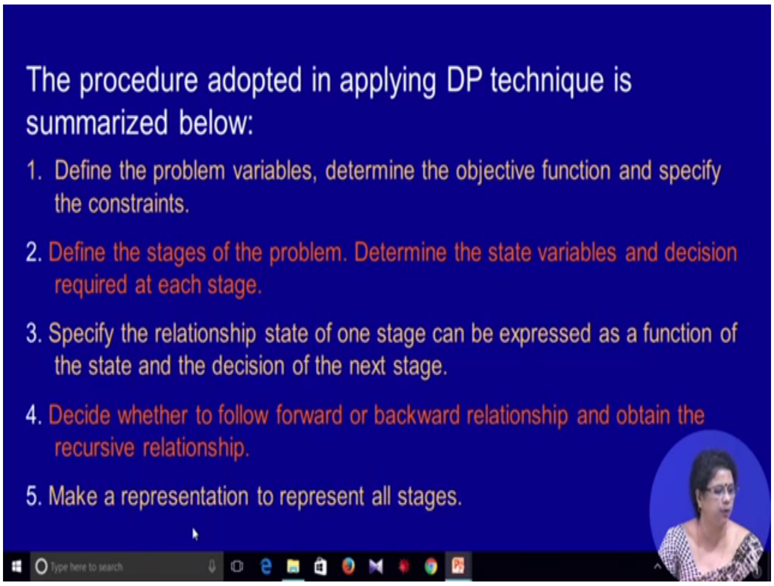
Now, you see we are having the recursion formula like this, here the objective function is of multiplicative form all right that is why at the inner stage.

If we defined in this way, f n x n is equal to maximum of f n x n that is the function with the n th value only and this is the F n minus 1, this is the optimal value for n minus 1 stage, that is why you see the Bellman's optimality principle says us that we did not too think about the initial condition, we are only concerned about whenever we are in 1 stage, we are concerned about the previous stage..

That is why at the n th stage, we are considering as input the optimal value of the n minus 1 stage and the objective function of the n stage is nothing else, since the problem is of maximization type and this is the multiplicative in nature that is why I can say return at the n th stage is equal to we have to maximize f n x n f n minus 1 s n minus 1. See the recursion here you start from n is equal to we can go in dynamic programming there are 2 ways we can do the recursion, we can go forward recursive formula we can use we can go from 1 to n, we can do the backward recursive formula as well from n to 1. we can do here we are going from 1 to n if we know 1 we can calculate 2 if we know 2 we can calculate 3 in this way we have proceeding, once we are calculating for n we need to know for n minus 1 that is the case.

Now, we will solve 1 non-linear programming problem with this recursive process and we will first calculate f 1 s 1 then f 2 s 2 then f 3 s 3 and we will reach to f n s n and every optimization problem would be problem of single variable, because you see here only the variable is s n no other variable is involving here is involved here all right. All other variables are not changing only s n is changing for different value of x n, which x n gives you the maximum return that would be the optimal at the n th stage, I hope you understood the basic principle of dynamic programming. Let us apply this problem for the non 1 specific non-linear programming problem.

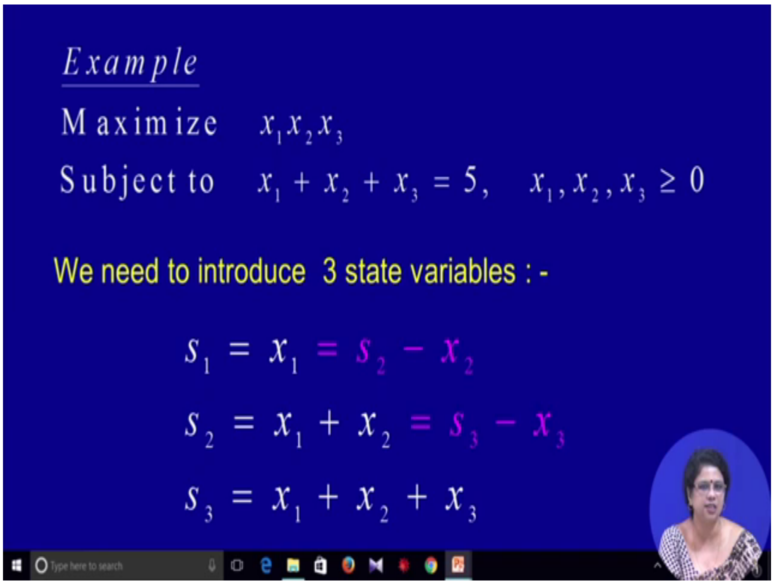
.(Refer Slide Time: 16:56)



Now, what is the procedure for doing the dynamic proof for applying the dynamic programming technique in non-linear programming problem, first of all we have we have to define the problem variables, we have to determine the objective function, we have to specify the constraint. Then we have to define the stage determine the state variables and corresponding decision at each stage, we need to find out this is the steps we need to follow one by one, once we got it after that we have to specify the relationship state of one stage has a function of state and the decision of the next stage.

If we go forward all right, now after doing this we will proceed we will follow either the forward recursive formula or backward recursive formula, to get the optimal solution at different stages these are all connected with each other, one optimal solution will be the of one stage would be the input of the next stage, that way we will proceed. It could be perform forward recursive, it could be backward recursion we will use it and we will solve the problem after that now.

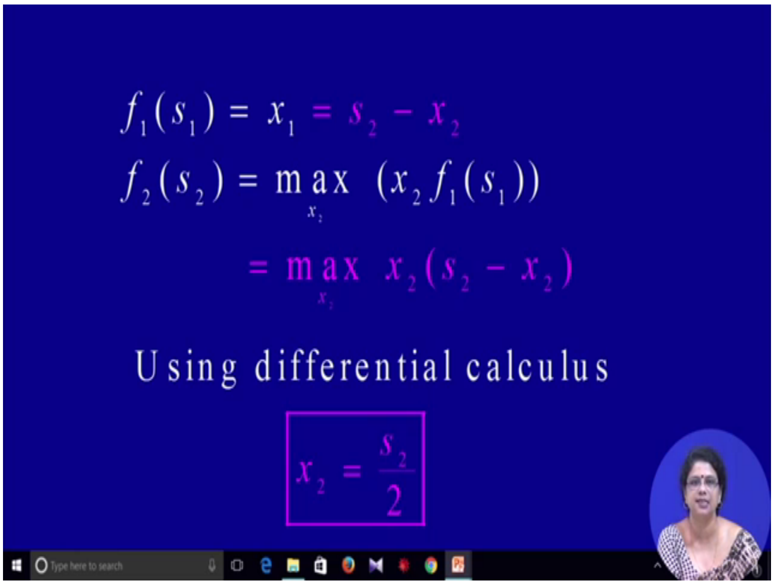
(Refer Slide Time: 18:27)



Let us consider a problem maximization of x 1 x 2 x 3 subject to x 1 plus x 2 plus x 3 is equal to 5, you see here the objective function is of multiplicative return and constraint is of additive type. Now we have to use 3 state variables like this by considering, s 1 is equal to x 1 s 2 is equal to x 1 plus x 2 s 3 is equal to x 1 plus x 2 plus x 3. We are just transforming from x 1 x 2 x 3 decision variable stages we are to s 1 s 2 s 3 combinations, you see we are getting a pattern s 1 is equal to s 2 minus x 2 s 2 is equal to s 3 minus x 3 in this way clear.

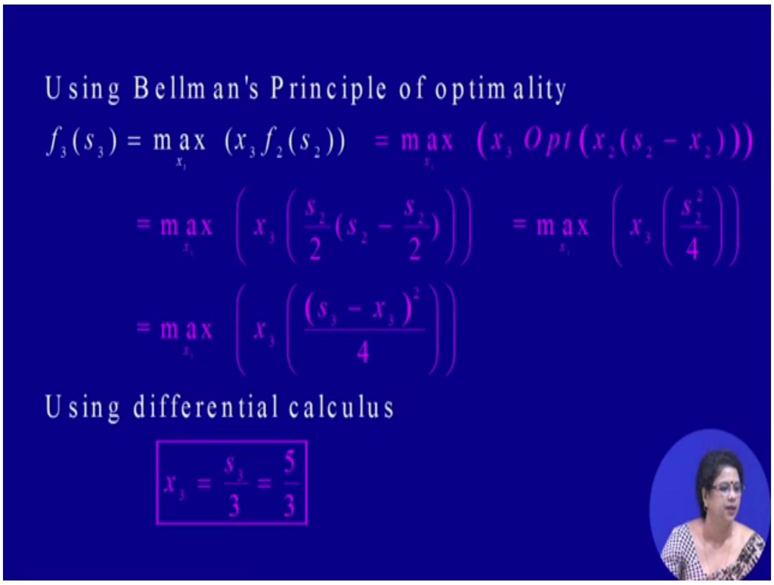
Now if this is so you see the first stage problem, will be only we are having the objective function is x 1 x 2 x 3 with the first state variable we can formulate the sub system as maximization of x 1, maximization of x 1 there is no constraint there is no end of it that is why no optimization can be done at this stage.

(Refer Slide Time: 19:26)



Let me proceed to the next stage we are going forward, that is the next step is f 2 s 2 this function how the function is the function is maximization over x 2 into x 2 and f 1 s 1 all right the previous. Now you see f 1 s 1 is equal to x 1, that can also be written as s 2 minus x 2 that is why this problem can be defined as maximization of x 2 into s 2 minus x 2, then this is a unconstrained problem of 1 variable that is x 2 and we will get the first necessary condition for it, if we just take the first derivative with respect to x 2 equate to 0, then we will get x 2 is equal to s 2 by 2 all right. We can go for the second order derivative as well to get whether this is gives you the maximum value or not, that also we can do with this value all right. This is quite clear that your function is s 2 into x 2 minus x 2 that is why if I just differentiate twice; we will get the value as minus 1 minus 1 means this gives you the maximum value. That is why you see the optimum value we are getting x 2 as s 2 by 2 ok.

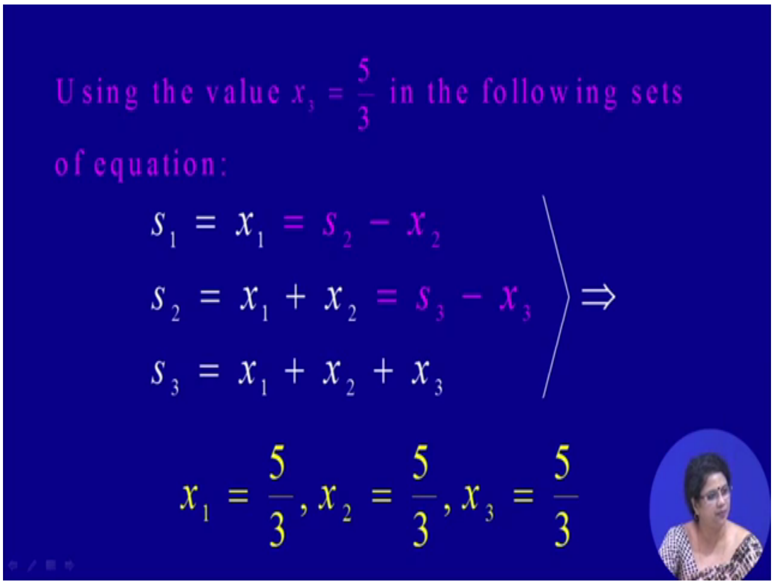
(Refer Slide Time: 21:29)



Let us move to the next stage that is the stage as x 3 rather s 3, if I just write down f 3 s 3 this is equal to max of over x 3 only x 3 f 2 s 2. Just now we got f 2 s 2 is a equal to what f 2 s 2 is equal to x 2 into s 1 all right is that clear, x 2 into s 1 we got and what is the value for x 2 x 2 value we got as s 2 by 2, x 1 can be written again as s 2 minus x 2 that is why x 1 is s 2 minus s 2 by 2 that is why that is also s 2 by 2 clear. That is why the optimum value up to the second stage we are getting s 2 square by 4 all right and we are having the function as maximization x 3 s 2 square by 4 this and maximization of x.

This is again one single dimensional problem, where they no constraint is there x 3 is the free variable it can take any positive value from 0 to infinity. Now again we cannot do it s 2 we know the relation that s 2 is equal to x 3 by s 3, that is why let us convert it into that form we will get it and from here by using the differential calculus method we will get x 3 is equal to x three by 3, what is your x 3 x 3 is equal to x 1 plus x 2 plus x 3 and you have the constraint that x 1 plus x 2 plus x 3 is equal to 5. That is why you are getting the value for x 3 is equal to 5 by 3. Once you are getting the value for x 3, you will get the value for which 1 s 2 how you will get the value for s 2 s 2 is equal to x 3 minus x 3, you will get the value for x 2 you will get the value for x 1 as well, that is why we can you see we can solve the problem very easily like this.

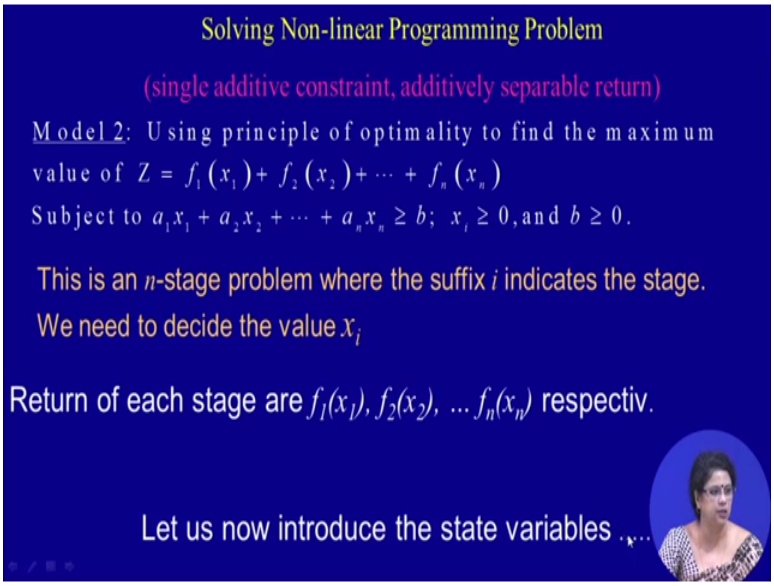
(Refer Slide Time: 23:39)



We will get x 1 is equal to 5 by 3, x 2 equal to 5 by 3 x 3 equal to 5 by 3 and the objective functional value would be 125 divided by 9 clear.

Now, we go we apply the Bellman's optimality principle for the case, where we were having the multiplicative objective function additive constraint.

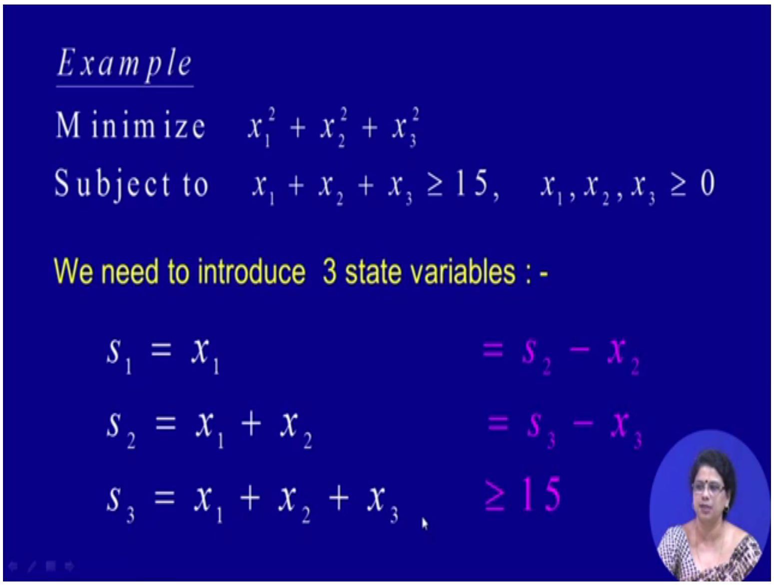
(Refer Slide Time: 24:06)



We can have the other way as single additive constrained, additive separable return and that is another model you see, how really we are using the states transformation equation. How really we are using the recursive formula here just to see first of all for this kind of problem, we have to define the state variables. How many stages we can define for this problem, we can have n number of stages similarly. Let us use the similar kind of state variables because constraints or of similar kind generally whenever using the non-linear, we are going to solve the non-linear programming problem using dynamic programming techniques.

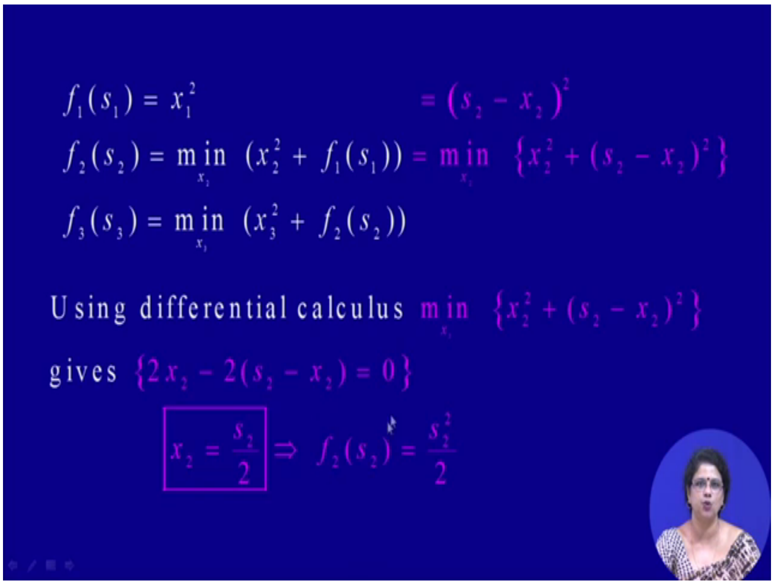
We are forming the state variables from the constraints state, only looking at the constraints state we have to decide what kind of constraint state we will use because constraint set will give you certain values, that is the only the restrictions we are getting in terms of numerals. That is why from there only we have to do it, now in the similar manner we will precede let me take one example for this.

(Refer Slide Time: 25:23)



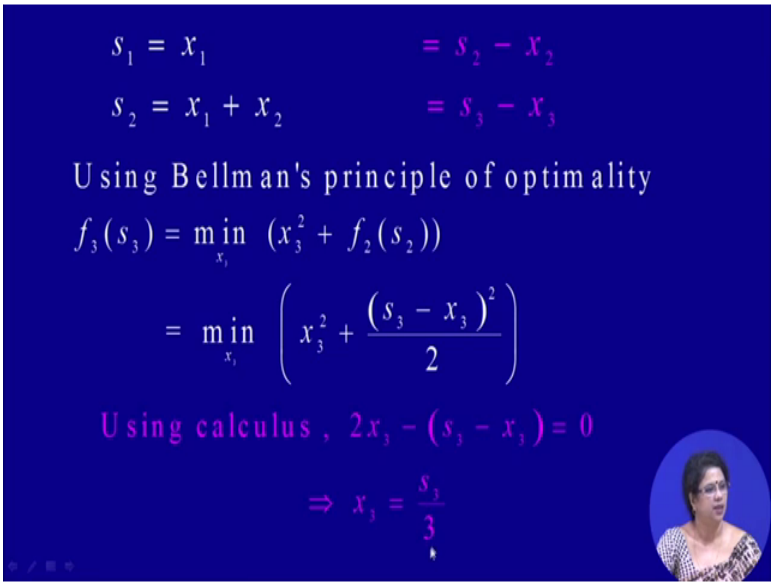
Just you see we are having the problem minimization of x 1 square plus x 2 square plus x 3 square, now you see the objective function is of type f 1 x 1 plus f 2 x 2 plus f 3 x 3, subject to a 1 x 1 plus a 2 x 2 plus a 3 x 3 greater than equal to 15 all right. Now you see summation is 3 is coming as 15 can you guess what could be the minimum value of x 1 square plus x 2 square plus x 3 square, try to guess it. Now we let me solve it, let me consider 3 state variables just like the previous 1 s 1 equal to x 1 s 2 is equal to x 1 plus x 2 s 3 is equal to x 1 plus x 2 plus x 3 and we know the fact that s 3 is greater than equal to 15 we know that fact and we have this relations as well.

(Refer Slide Time: 26:33)



Let us formulate the transformation equation let me go from the first stage f 1 s 1 x 1 square f 2 s 2 minimum of over x 2 x 2 square plus f 1 s 1, let me convert x 1 as s 2 minus x 2 that is why minimization will be done this way. Similarly for 3 if we just use the differential calculus method again, we are getting x 2 is equal to s 2 by 2 you can check it whether this is correct or not correct and the corresponding functional value we are getting s 2 square by 4 because this is x 2 square it is not 2 it is 4 clear.

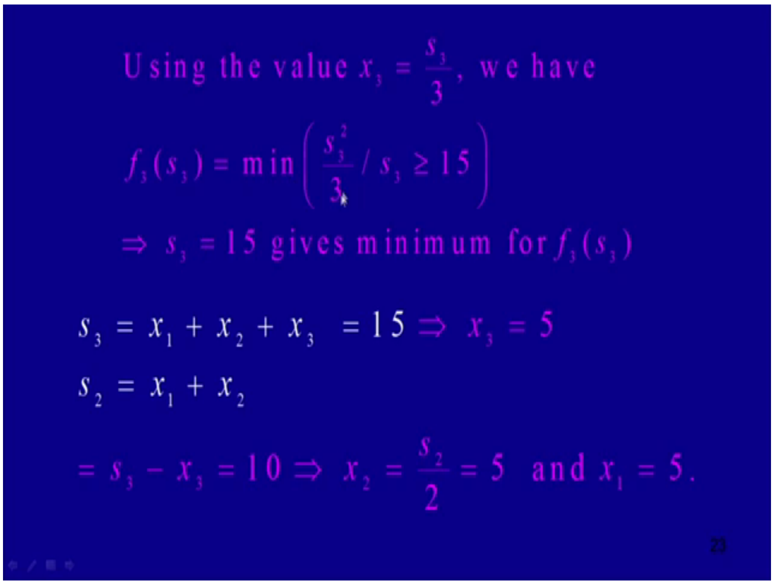
(Refer Slide Time: 27:14)



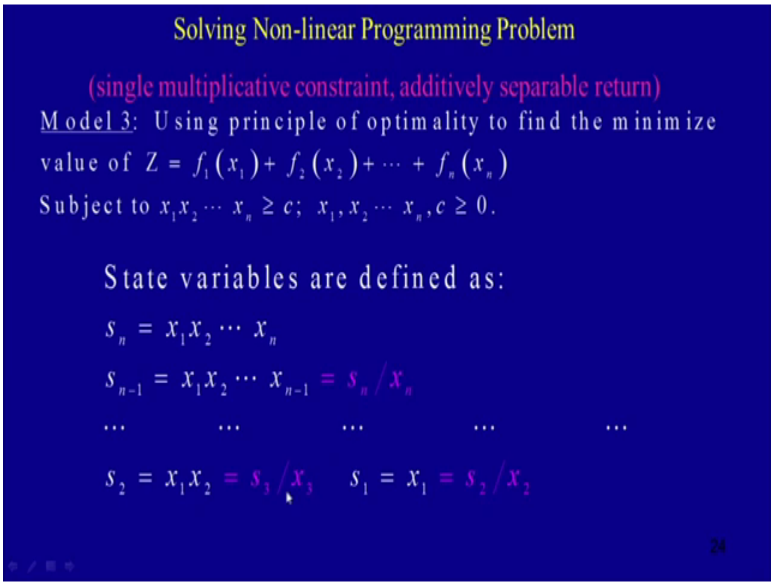
Now, we are moving to the third stage, now we know this fact all right this is 4 we know this fact if we are getting x 3 is equal to s 3 by 3 all right.

What is the minimum value for s 3 minimum values for x 3 is equal to 15 we know from there we can get that f 3 s 3 minimum of these certainly s 3 is equal to fifteen will be the will be the minimum value all right. If s 3 is 15 then we will get s 2 is equal to and for which value of x, x 3 that is x 3 is equal to s 3 by 3, that is why x 3 would be is equal to 5 if you get x 3 equal to 5 then you will get x 1 plus x 2 is equal to this 10 and x 2 is equal to 5 x 1 equal to 5. That is the minimum you must have cased before this way we can solve the problem.

(Refer Slide Time: 28:19)

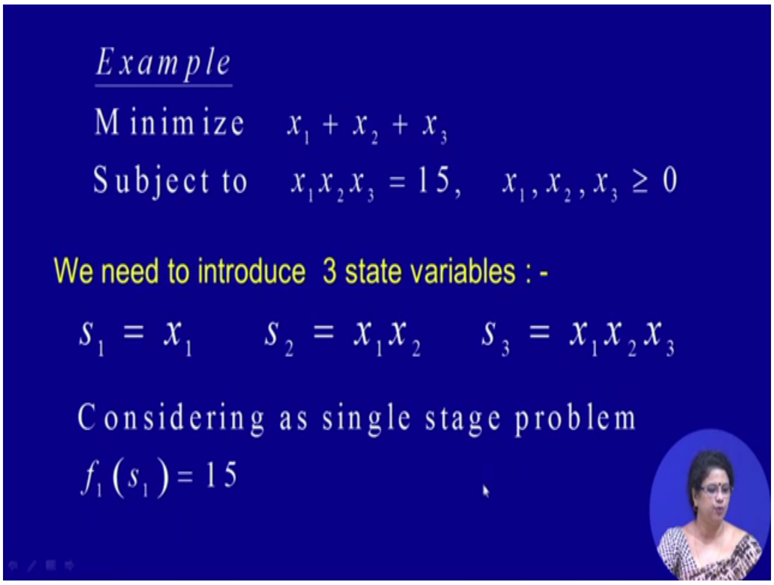


(Refer Slide Time: 28:20)



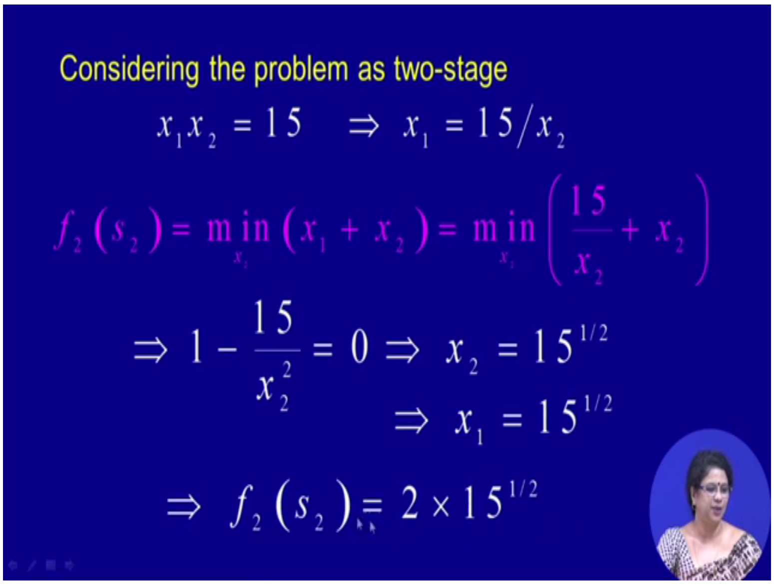
Let me consider another model, you see we are having the additive return, but multiplicative constraint then what kind of state variable will consider in this case, they will just as I say to you that looking at the constraints set we have to decide about the nature of the state variables. That is why the state variable should be s n would be the multiplication of this s n minus 1, would be this and s 2 would be this and s 1 will be this 1. Only in the changes there you see the pattern is changing instead of minus, here division because the constraint is off multiplicative type all right.

(Refer Slide Time: 29:03)



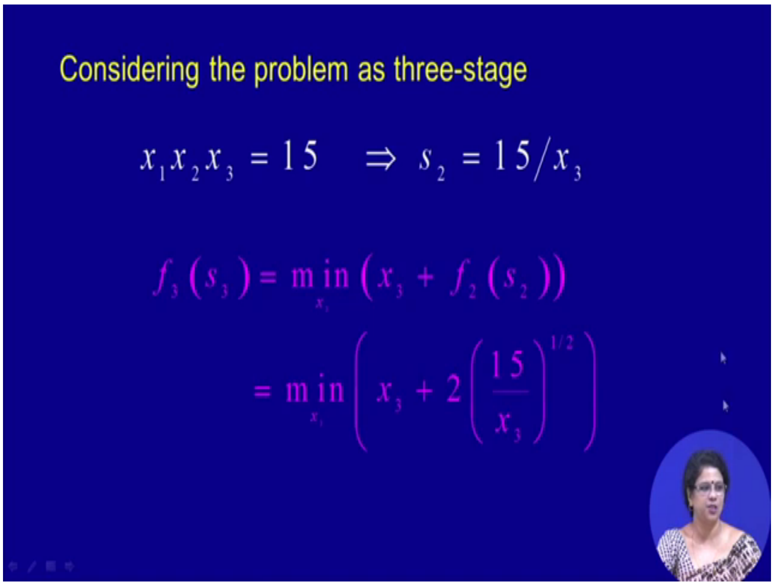
We let us solve the problem again like this, we will let me use the state variables let me solve we will get the solution.

(Refer Slide Time: 29:13)



We will see that x 1 will be is equal to this, then x 2 would be is equal to this 1 if it is use the which 1 the differential calculus approach over x 2. Then we can get the value of x 2 is equal to fifteen root of 15 now if x 2 is root of 15, then x 1 would be again root of fifteen then we will get the value for x 3 as well, but we need to proceed for the third iteration otherwise we cannot.

(Refer Slide Time: 29:49)



Complete the process that is why we are moving to the next iteration that is s 3 all right s 3 is this 1, again use the differential calculus approach and we will get the value for x 3 is equal to15 to the power 1 by 3. Now we are getting the objective functional value you see 3 into 15 to the power 1 by 3 from here all right and what about ha what about the value for x 2 value for x 2 was 15 to the power 1 by 2 all right and x 1 is 15 to the power. Just you check whether we are getting the same satisfy the same constraint or not the multiplication is coming as how much is coming.

Student: (Refer Time: 30.50)

That is why individually it would be 1 by 3.

Student: (Refer Time: 31.01).

Ah?

Student: X 3 should 1.

X 3 should 1; you just do the calculation once more. I will give you assignments on this part and we will give you the solution as well for this problem, now try to solve this problem using the nth stage dynamic programming technique.

Thank you for today.