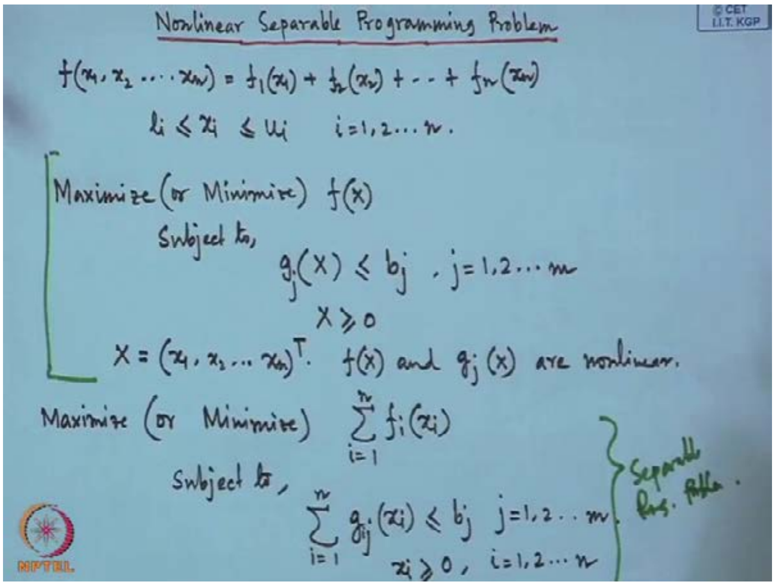
**Optimization**
**Prof. Debjani Chakraborty**
**Department of Mathematics**
**Indian Institute of Technology, Kharagpur**

**Lecture - 34**
**Separable Programming Problem**

Today's topic is nonlinear separable programming problem. Now, separable programming problems are special kind of nonlinear programming problems in which the nonlinear objective functions and the functions involved in the constraints are separable in nature. Separability means that, if a function is separable, then the function can be written as an addition of several terms, where the individual terms are the functions of single decision variables.

(Refer Slide Time: 00:55)

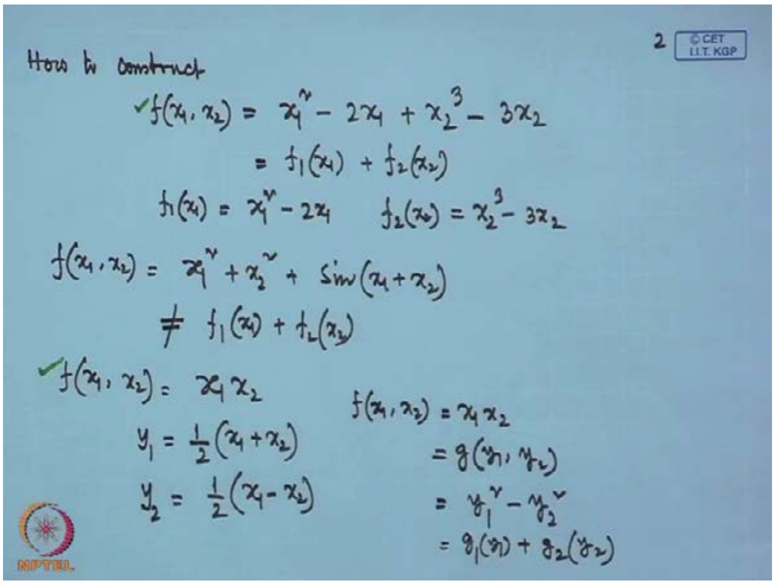



Mathematically, we can say that, if a function f of decision variable x 1, x 2, x n is said to separable if it can be written as f 1 x 1 plus f 2 x 2 – sum of n number of functions of individual decision variables. Here I need to say that, a function f is nonlinear function. The functions f 1, that is, a function of x 1; f 2 – the function of x 2 – these are the functions; it could be linear; it could be nonlinear as well. Now, the way we are handling the separable programming problem; the basic principal is that, we just express the nonlinear functions f i x i as the combinations of linear functions, so that a nonlinear programming problem will be transformed into a linear programming problem. And, for solving the linear programming problem, we have the simplex algorithm. And, with the restricted conditions, we can solve the simplex algorithm for handling the separable programming problem. Now, this is the basic idea. One thing is needed for separable programming problem solution that, for individual variables, that is, decision variable x I; we should have the lower bound and the upper bound; where, i is equal to 1 to n.

Now, let us consider a general nonlinear programming problem, that is, maximize or minimize f X subject to g j X less than is equal to b j say, where there are n number of constraints are there. And, X is the decision variable and X could be a vector with n components. Now, f X at g j X are the nonlinear functions. Now, as I said that, for the separable programming problem, the objective function f X and the constraint functions, that is, g j X – these are separable in nature. Therefore, the same problem – the nonlinear programming problem can be written in separable programming problem in the way that, these maximize or minimize… f X is the function of n number of variables, which can be written as the separable functions. That is how we can write summation of i is equal to 1 to n f i x i; where, small x i is the i-th decision variable. And, the constraints would be summation i is equal to 1 to n. m number of constraints are there. That is why g i j x i less than is equal to b j; where, j is equal to 1 to m. And, here x i's are all greater than equal to 0, that is, a non-negativity constraints; that is, for i is equal to 1 to n.

Then, this is the format of a general separable programming problem. This is the general separable programming problem. Now, how to handle this problem? Now, for solving this problem, what we do; as I said, the basic principle is that, we will just express individual nonlinear functions, that is, f i's and g i j's as the combination of the linear functions. Thus, we need to know that; two things we need to know. One is that the – when a function will be said as the separable function; or otherwise, how function can be expressed in the separable form if it is possible. And, the second thing is that, how a nonlinear function will be expressed as the combination of the linear functions. That is why let me detail both the things one by one.

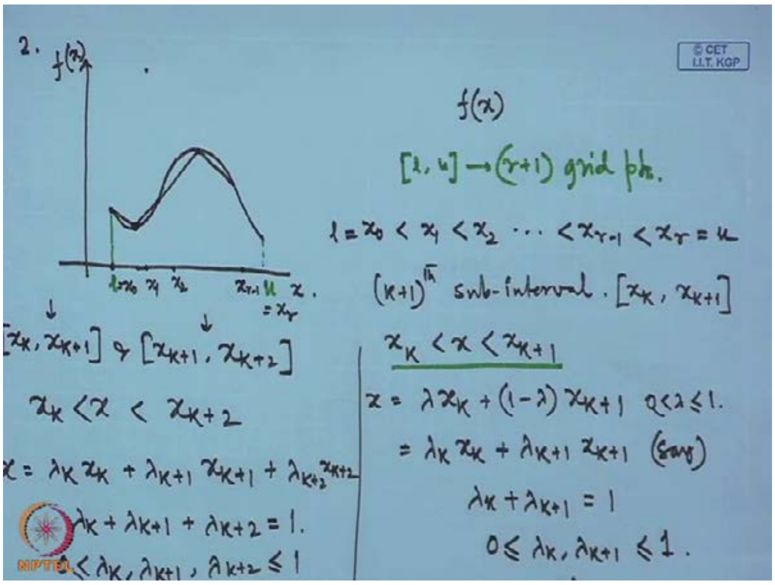
(Refer Slide Time: 05:49)




The first is that, how to construct a separable function. Now, as I said that, let me consider one function of two variables: f x 1 x 2. This is equal to x 1 square minus 2 x 1 say x 2 cube minus 3 x 2. Now, this is the function of two variables. Just see if we can write down this function in the form f 1 x 1 plus f 2 x 2 very easily, because it could be… We can separate the function say f i's for the function of x i's; where, f 1 x 1 is equal to x 1 square minus 2 x 1; and, f 2 x 2 is equal to x 2 cube minus 3 x 2. But, if we consider a function f x 1, x 2 as x 1 square plus x 2 square plus sine x 1 plus x 2; we cannot write this function in the form of f 1 x 1 plus f 2 x 2, because of the involvement of sine x 1 plus x 2. That is why this function is not a separable function. That is why separable programming technique is not applicable for such kind of function involvement.

Let me consider another function say f x 1, x 2 is equal to x 1 x 2 – the multiplication of x 1 and x 2. Though it cannot be directly written in the form of f 1 x 1 plus f 2 x 2; but, if we simplify some other way; then, we can write as a summation of separable function. Just see. Let me consider one variable y 1 as half of x 1 plus x 2; and, let me consider another variable y 2 as half of x 1 minus x 2. If we just write down in this way, then very easily we can write down f x 1 x 2, that is, given to us; x 1 x 2 as a separable function of y 1 y 2; how? Simply we will write y 1 square minus y 2 square. Then, that would be x 1 x 2. And, certainly this is the function of y 1 plus function of y 2. Thus, this function is the separable function. Thus, we can conclude that, the functions, which are nonlinear functions, which are involved in the objective function or in the constraint if this is in the form of separable function; then only this – the methodology, which I am going to describe, is applicable for such kind of problem.

(Refer Slide Time: 09:04)



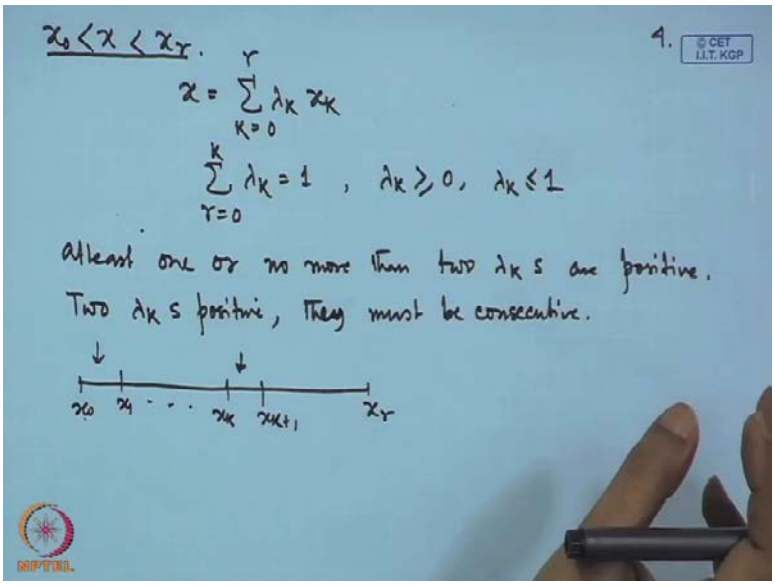
Now, the next – how to handle? How to convert the nonlinear function into the separable form, that is, into the combination of the linear function; that part I am going to tell you. Let me consider a function like this. For this thing, I will consider only the single variable function say f x, because then only I can draw the function easily. The same idea can be extended for n number of variables x and this is the function f x. Now, if I consider this function in this way; now, as I said for the separable function; then, for the separable function, the lower limit say l and upper limit say u is given to us. Then, just see if we can express in nonlinear function in a combination of linear functions. Let me just describe geometrically first. Then, I will go for the mathematical expression for this.

Now, let me partition this interval – l, u interval with r plus 1 number of grid points. How? This l would be is equal to x 0; x 1 is the next; x 2 is the next point. In this way if I just proceed, u would be is equal to x r and this is the previous point x r minus 1. Now, how we can express the c? We will consider the point x naught; we will consider the point x 1. In this interval, we will just add f x naught and f x 1. Then, we will get a straight line. Thus, I can say within this interval; then, function can be expressed as a linear function instead of a nonlinear function. In the interval x 1 and x 2, this can be written; the function can be expressed in this way. Similarly, for the next interval, function can be expressed. And, this is very clear that, if I consider the combinations of all such functions together; then, that can approximate the nonlinear function f x very easily. And, this is to be noted that, if the number of grid points are more; rather if the points are very close to each other; then, the approximation we are making; that approximation will be the best approximation.

Now, how mathematically we can express these things together? Just let me see. How I have written the grid points? I have written the grid points as x naught lesser than x 1, lesser than x 2; in this way, x r minus 1, x r; that means r number of partitions we are having and l is the lower limit and u is the upper limit of this function; that means the function f x for the decision variable, not for the function. Thus, the function f x is defined within the range l to u. Now, if this is so, let me consider the k plus 1-th are subinterval. Now, any point x… k plus 1 is subinterval means the interval x k, x k plus 1. Any point x in between x k and x k plus 1 can be expressed as the linear combination of x k and x k plus 1. Now, where we know if we consider the linear combination, lambda must be in between 0 to 1. Now, I can write down this expression as lambda k x k plus lambda k plus 1 x k plus 1; where, I want to introduce the lambda coefficient with the k index, because I will extend this idea further. Just I will explain in the next. Just see what is the condition here. Lambda k plus lambda k plus 1 must be is equal to 1. And, what is the next condition? Lambda k and lambda k plus 1 – both are in between 0 to 1. Thus, if we have one interval x k to x k plus 1; any point x can be expressed as lambda k x k plus lambda k plus 1 x k plus 1.

Now, let me extend this idea for the three intervals; that means I will consider the interval x k, x k plus 1 – three grid points; and, the next interval as well – x k plus 1, x k plus 2. Then, any point in between x k to x k plus 2 can be written as lambda k x k plus lambda k plus 1 x k plus 1 plus lambda k plus 2 x k plus 2 with the condition lambda k plus lambda k plus 1 plus lambda k plus 2 is equal to 1; and, with another condition that, lambda k, lambda k plus 1, lambda k plus 2 – these are all in between 0 to 1. Now, once we are expressing x, which is in between x k to x k plus 2; if it is can be written in this way; then, certainly either x is within the interval here or x is here. That is why lambda k, lambda k plus 1 and lambda k plus 2 – all cannot be simultaneously in the positive level; either if x in between x k to x k plus 1, we will have lambda k and lambda k plus 1 positive; or, if x is in the interval x k plus 1 and x k plus 2, then lambda k plus 1, lambda k plus 2 is positive. But, here lambda k plus 2 is 0; and, here lambda k is 0, because x cannot lie simultaneously in both the intervals. I am expressing x in terms of three grid points: x k, x k plus 1 and x k plus 2. What is my objective? My objective is to express the function f x in terms of r number of linear segments – r number of linear functions. That is why first let me analyze the point – the x, which are lying in the respective partitions.

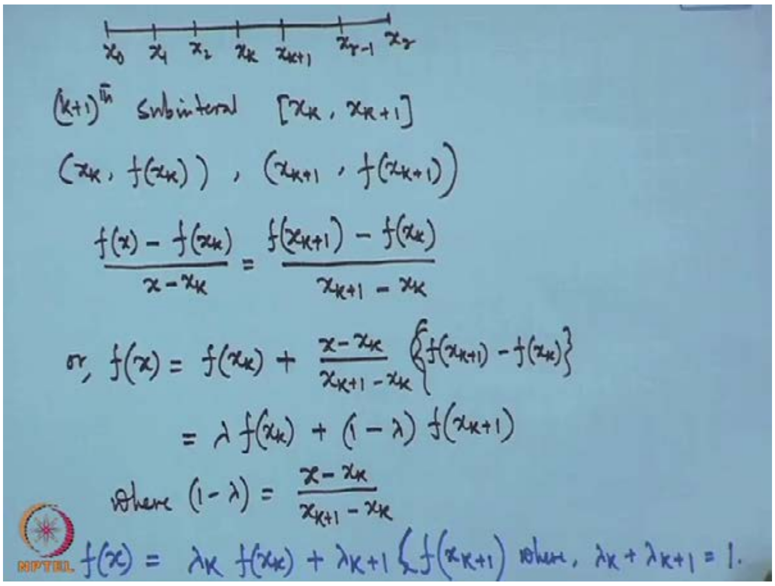
(Refer Slide Time: 16:29)




Now, we have considered two points; we have considered three points here. If I extend this idea further, then for r plus 1 number of grid points; that is, if we consider all grid points when x is lying in between x naught to x r; any x can be written as x is equal to lambda 0 x 0 plus lambda 1 x 1 plus lambda 2 x 2. In this way, I can write it is lambda k x k; k is in between 0 to r; where, summation of r is equal to k; lambda k is equal to 1. Not only that; individual lambda k's are greater than equal to 0 and lambda k's are less than as well equal to 1. And, another condition we should impose that, since x is a point in between x naught to x r; x must be in either of the partitions. That is why x cannot be simultaneously lying in two partitions at a time. That is why we should impose the condition here at least one or no more than two lambda k's are positive.

Not only that; if two lambda k's are positive, then they must be consecutive, because we are considering the partition x 0, x 1, x k, x k plus 1 in this way up to x r. If x is here, then lambda k and lambda k plus 1 will be positive. If x is here, then lambda 0 and lambda 1 will be positive; other will be 0. In this way we can write down the function; we can write down the decision variable x in terms of r plus 1 number of grid points. The similar idea – the same idea we will extend to express the function f x in terms of k r plus 1 number of grid points. Just see in the next how I am expressing that fact.

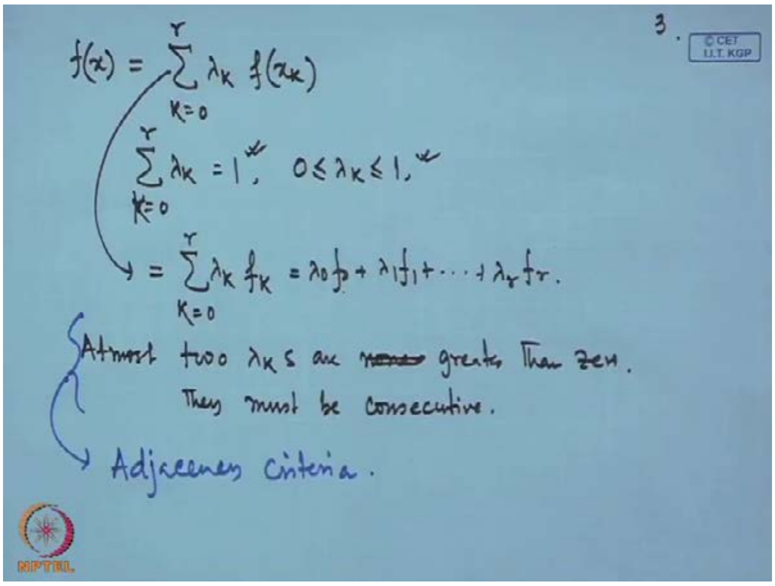
(Refer Slide Time: 18:57)




Let me consider again the whole interval, that is, l to u; where, the function f x is defined. And, we are considering the grid points x 1, x 2, x r minus 1, etcetera. Now, again we are considering the k plus 1-th subinterval; that is, x k to x k plus 1. Say this is the function for us. This is x k; this is x k plus 1; and, this is the function here. If this is the function here; I am expressing the function in this interval as a linear function. That is why what we will do; in this interval, we will just draw a line between two points: x k, f x k; x k plus 1, f x k plus 1. What would be the equation of the line? In other way, we can write it as f x is equal to f x k plus x minus x k divided by x k plus 1 minus x k f x k plus 1 minus f x k. Just I have simplified the previous equation. In other way, we can write it lambda into f x k plus 1 minus lambda into f x k plus 1; where, 1 minus lambda is equal to x minus x k divided by x k plus 1 minus x k. Thus, we see that, any f x within the interval x k and x k plus 1 can be written as lambda k f x k plus lambda k plus 1 f x k plus 1; where, certainly lambda k plus lambda k plus 1 is equal to 1, because we have just considered lambda k plus 1 as this one – x minus x k divided by x k plus 1 minus x k. And, lambda k would be is equal to 1 minus of this amount. Thus, we can say that, lambda k plus lambda k plus 1 is equal to 1.

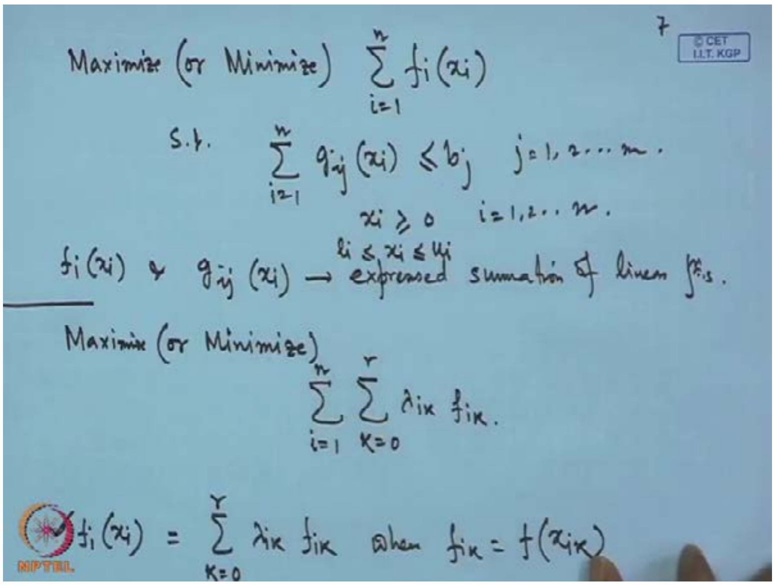
Thus as we see that, any function f x can be expressed as the linear function here. What is f x k? f x k is the functional value at the grid point x k. f x, that is, the constraint value certainly. f x k plus 1 is the functional value at the grid point x k plus 1. And, here lambda k and lambda k plus 1 – these are the unknown variables for us. Thus, we can say f x is being expressed as the linear function very nicely. Just we have considered one subinterval. If we extend this one just like as we did for x; we have extended that for three grid points and for r plus 1 number of grid points as we have done; the similar way we can do it for the r plus 1 number of grid points.

(Refer Slide Time: 23:31)




And, we can write down the function f x as summation lambda k f x k; k is equal to 0 to r; where, summation of lambda k, that is, from k – 0 to r is equal to 1; individual lambda k's are all in between 0 to 1. And, another condition is also there. Now, this can be written as summation k is equal to 0 to r; lambda k f k; rather this is equal to lambda 0 f 0 plus lambda 1 f 1 in this way – lambda r f r. Just see what we have done; we have again considered the functional values at the grid points x 0, x 1 up to x r; r plus 1 number of grid points. We have introduced the variables ((Refer Slide Time: 24:40)) plus 1 number of variables; lambda 0, lambda 1 to lambda r in such a way that, these two conditions hold together. Summation of lambda k is equal to 1 and the individual lambda k's are in between 0 and 1. And, again the same condition we have to impose here that, at most two lambda k's are nonnegative, rather greater than 0. And, if so, they must be consecutive. Thus, this is the basic principle for handling the separable programming problem; that is, any function, which can be written in the forms – separable form; individual functions will just express in this way as the combination of the linear functions; and, we will handle in the next. Now, what is the advantage in this procedure? And, one thing I forgot to mention that, this criterion is called the adjacency criteria, because this criterion is very much important when we will apply the simplex algorithm in the next.

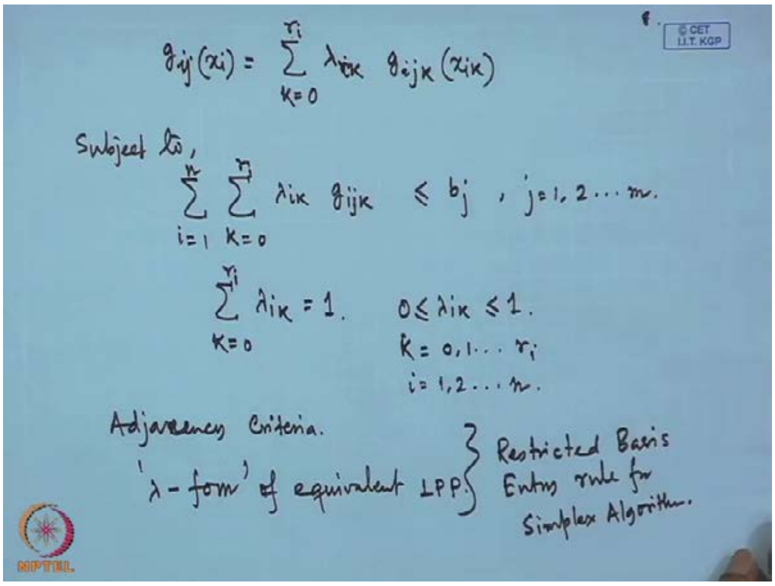
(Refer Slide Time: 26:29)




Now, coming back to the separable programming problem; now, what is that? Maximize or minimize summation f i x I; i is equal to 1 to n; n number of decision variables are involved such that, i is equal to 1 to n g i j x i less than is equal to b j. There are m number of constraints are there. And, x i's are all greater than equal to 0; i is equal to 1 to n. Now, by applying the previous procedure, each f i x i and g i j x i; these are the nonlinear functions. And, these are the functions of individual decision variables. Thus, this can be expressed as combination of summation of linear functions just like before. Thus, this separable programming problem… And, one thing is also there that, individual decision variables are bounded; otherwise, separable programming problem technique is difficult to apply, because we cannot partition the interval.

Now, this can be written nicely in this fashion, that is, maximize or minimize. Summation i is equal to 1 to n. Just see how I will write f i x I; f i x i can be written as… I am introducing another index k here. There are r plus 1 number of grid points. That is why f i x i can be written as lambda i k f i k. Thus, we can write down here i is equal to 1 to n; k is equal to 0 to r lambda i k f i k; where, f i k is equal to functional value at the grid point x i k. x i is the decision variable; and, x i k is the k plus 1-th grid point. Thus, in the separable programming problem, the objective function can be expressed as the summation of the linear functions. This is a linear function. Thus, f 1 x 1 is a linear function; f 2 x 2 is a linear function in this way. We will get a summation of individual terms, where the terms are all the linear functions. Now, this is this above; this is above the objective function. Let me come to the constraint. That is how I will express g i j x i as the linear functions in the next.

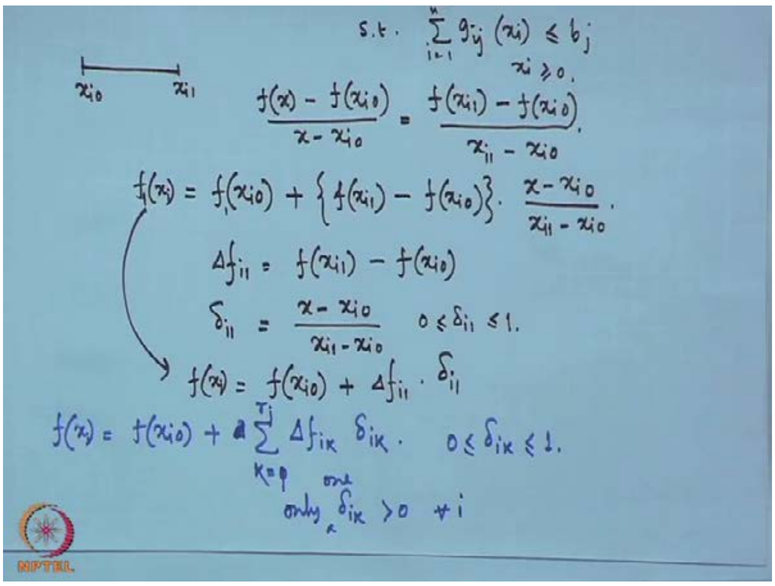
(Refer Slide Time: 29:52)




This is the nonlinear function. Again this can be written as the combination of k plus 1 terms, where individual terms are the linear terms; lambda i k g i j k and x i k. If this can be written as this, then certainly I can write down the constraint set as for the separable programming as summation k is equal to 0 to r. And, there is another summation also is there; i is equal to 1 to n and lambda i k g i j k. And, this is less than is equal to b j; where, j is equal to 1 to m. Not only that; there is another condition should be there; that is, k is equal to 0 to r; lambda i k must be is equal to 1; individual lambda i k's are 0 to 1; where, i k is equal to one thing. I should mention here that, here you see r x i for the decision variable x i; the grid points are x 0 i, x i 0 x, x i 0 x i 1, x i r. Thus, r i. Thus, this is the number of decision… Number of partitions are say r i, because for individual decision variables, we can have partitions with different numbers; not only that, I should mention here one thing is that, the partition may not be the equally spaced; it could be unequally spaced as well depending on the nature of the function. Thus, instead of partitioning the range of x i… We are partitioning with r i plus 1 number of grid points. Grid points are there; thus, here as well the… Here this is i; this is i; thus, this is i. And, here k is equal to 0, 1 to r i and i is equal to from 1 to n, because n number of decision variables are there.

This is the equivalent form linear programming problem of the original separable programming problem. And, this can be solved with the simplex algorithm and imposing the adjacency criterion; otherwise, the linear combinations we have considered – that is not useful. Now, this form; since we have introduced the lambda decision variables here; this form is called the lambda form – lambda form of equivalent LPP. And, this adjacency criteria means that, in this linear programming problem, when we will solve it with simplex algorithm; there will be a restricted basis entry rule, will be imposed; and, that is that, at a time, 2 lambda i k's can be… At the most, 2 lambda i k's can be in the positive level. And, that is, I can write down that, this is the restricted basis entry rule for simplex algorithm. I will explain this further with numerical example that… Then, you can understand what I mean to say about the restricted basis entry rule. Now, similarly, instead of lambda form, we can have some other two forms as well for expressing a nonlinear separable programming problem with an equivalent linear programming problem. I am coming to that in the next.

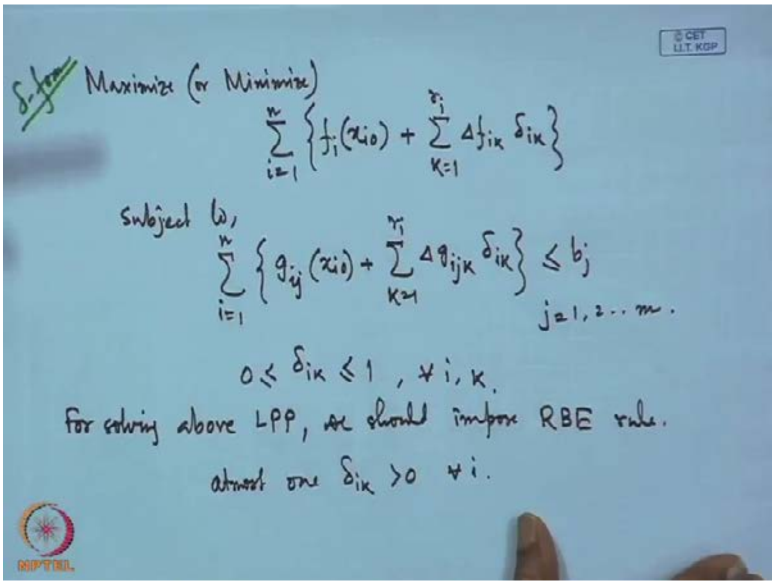
(Refer Slide Time: 35:09)



One form is the del form; and, another one is there, that is, the delta form; that I am coming to the next. We are having the separable programming. Again consider the separable – original separable programming in this way; f i x i i is equal to 1 to n subject to i is equal to 1 to n g i j x i less than is equal to b j. See we have considered here the approximations in this way. Let me consider the first interval; x i 0, x i 1. How f x is being written here? I can write down f x minus f x i 0 divided by x minus x i 0 is equal to f x i 1 minus f x i 0 divided by x i 1 minus x i 0. Rather I can write down f x as equal to f x i 0 plus f x i 1 minus f x i 0 into x minus x i 0 divided by x i 1 minus x i 0. Let me consider del f i 1 is equal to f x i 1 minus f x i 0; delta f i 1; and, del i 1 as x minus x i 0 divided by x i 1 minus x i 0. If I am writing so, then f x can be written as f x is equal to f x i 0 plus del f i 1, that is, a constraint, because we considering the difference between the functional values in the consecutive grid points into del i 1; f x i; f x i. Now, if this is so; now, you see here this delta i 1 is in between 0 to 1. Thus, you see as in the lambda form, we have introduced the lambda decision variable. And, here we can instead of introducing lambda decision variable, we can introduce the decision variable del i 1.

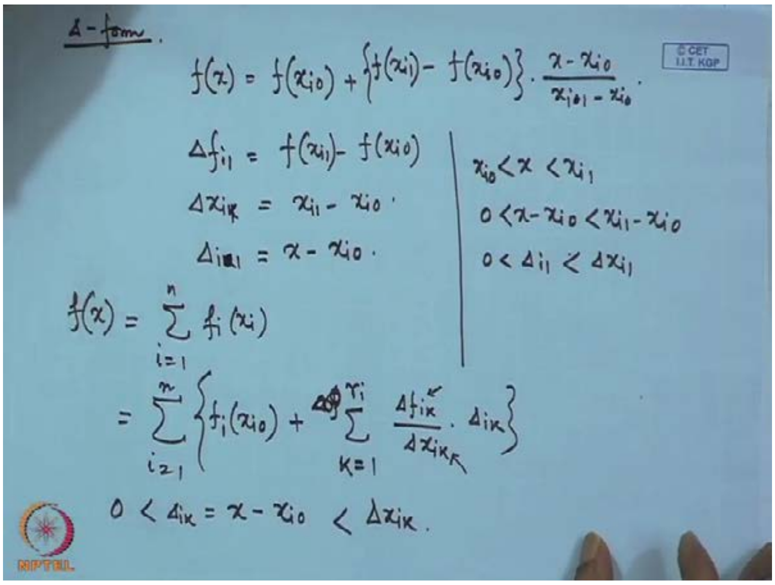
Similarly, for r plus 1 number of grid points, this can be generalized as f x is equal to f x i certainly – f x i 0 plus del summation k is equal to 1 to r i del f i k del i k. And, here delta i k less than is equal to 1. Now, instead of going for the lambda form, we can use the del form as well. One advantage here is that, only one rule I can impose here that, only one delta i k is positive at a time, because x can be in any one of the subintervals and we have considered the proportion of the subinterval. And thus, we can say this is the difference of the functional value in the consecutive grid points. Thus, while we will consider the linear programming problem; in that time, the restricted basis entry rule we have to impose in the del form is that, only one del i k will be in the positive level for all i. With this idea, let me express the separable programming problem once more. Thus, I can write down the separable programming problem in this way.

(Refer Slide Time: 40:28)




Maximize or minimize summation; individual functions can be written in this way – f x i 0 plus k is equal to 1 to r i del f i k. And, since there are n number of decision variables, this can be written as i is equal to 1 to n base. And, subject to the condition, i is equal to 1 to n and individual constraints function in the similar way I can write down g i j x i 0 plus summation k is equal to 1 to r i del g i j k. We will consider the constraint function as the individual grid points; and, this is the del i k. This must be less than is equal to b j; j is equal to 1 to m. Here del i k is in between 0 to 1 for all i and k. And, for solving above linear programming problem, we should impose restricted basis entry rule as at most one del i k must be positive for all i. Thus, this is that we have achieved to the del form of the delta equivalent form of separable programming problem. Now, there is another form, that is, delta form.

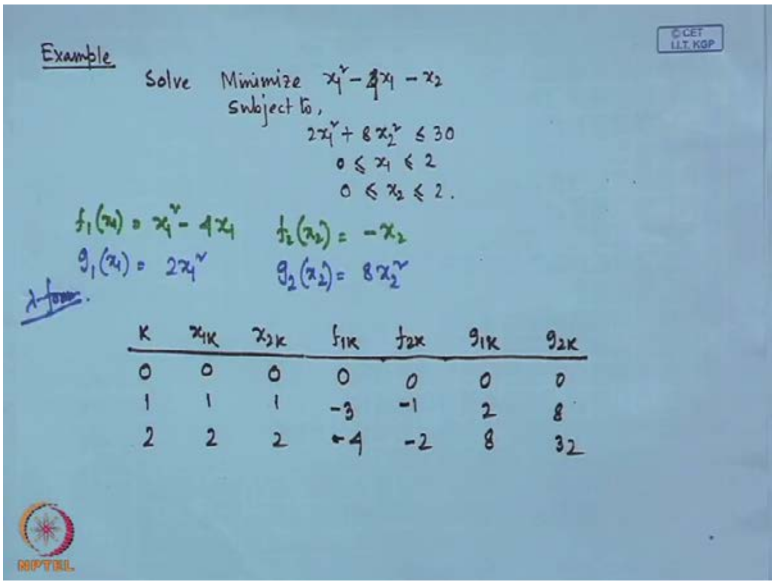
(Refer Slide Time: 42:56)



Let me write down that form as well; that is, delta form. Here the consideration is little bit different; other things are same. As I have considered that, this f x as f x i 0 plus f x i 1 minus f x i 0 x minus x i 0 divided by x i 1 minus x i 0. Here we are considering some other decision variables. We are considering del f i 1 as f x i 1 minus f x i 0. And, we are considering the difference of x as well; f x i 1 as x i 1 minus x i 0. And, we are considering del i k as x minus x i 0. That is the only consideration here, so that f x can be written as… Let me write down in general form; any f x can be written as summation of i is equal to 1 to n f i x i. This is equal to summation i is equal to 1 to n f i x i 0 plus delta. There will be summation k is equal to 1 to r i delta f i k divided by delta x i k into delta i k. One thing you just see; this is the constant here, because we are considering the difference between the functional value at the k plus 1-th grid point and the k-th grid point; this is the… We are considering the difference of the grid points only. This is also constant. Only the decision variable is del i k. And, what is the value for del i k? del i k can be… This is the value for del i k. Just see the range for del i k.

As we know that, in this case, x is in between the first interval x i 0, x i 1. That is why x minus x i 0 must be in between x i 1 minus x i 0. Thus, I can say that, del i 1 – here not k, here the index is 1; del i 1 must be in between 0 to del x i 1. Thus, I can say here del i k in between 0 to del x i k. Thus, you see this is the objective function. Similarly, we can write down the constraint function as well; and where, the decision variables will be delta i k's. And, this is in between 0 to del x i k. And, these are all the constants. Now, whatever theory just we got; the same thing we will just apply for the numerical example in the next.

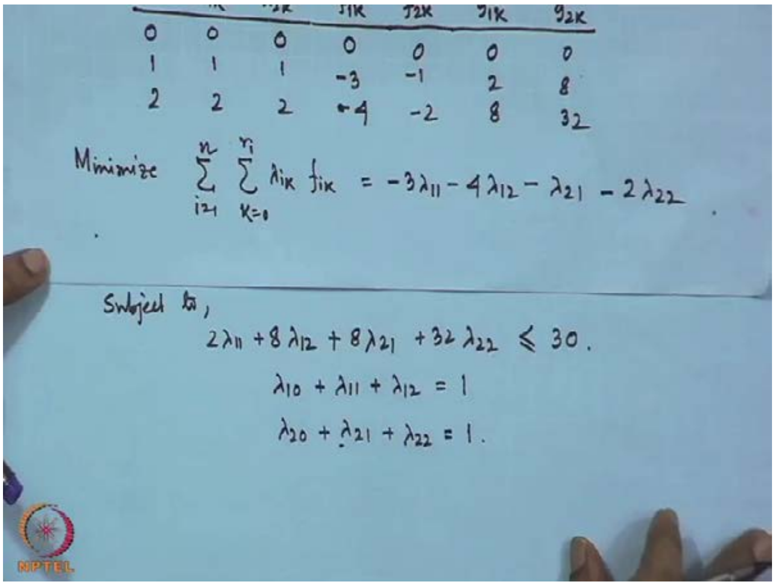
(Refer Slide Time: 46:39)




Let me consider this numerical example; that is, the function of two variables minimize x 1 square minus 4 x 1 minus x 2 and subject to the nonlinear constraint – 2 x 1 square plus 8 x 2 square less than is equal to 30. And, x 1 is in between 0 to 2 as well x 2. Now, this is a separable; we can applied a separable programming technique, because you see the function f 1 x 1 is equal to here x 1 square minus 4 x 1; f 2 x 2 is equal to minus x 2. Similarly, for the constraints, g 1 x 1 is equal to 2 x 1 square and g 2 x 2 is equal to 8 x 2 square. And, these are the bounds of x 1, x 2. Now, for handling this thing, either we can apply the lambda equivalent form, which is LPP; we can use the del equivalent form. In any one of these three, we can apply.

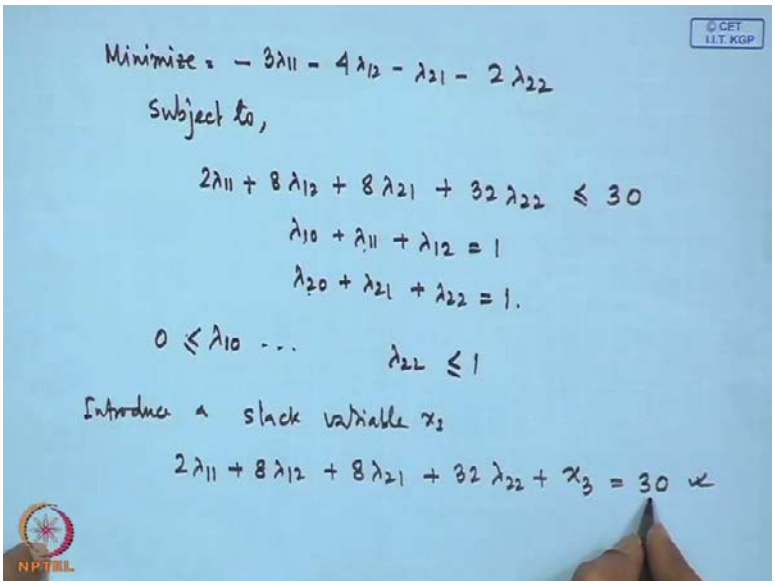
Let me go for the lambda form. Now, for doing this thing, we need to find out the functional values f i's and g i j's at the grid points. Now, since the x 1 is in between 0 to 2, I can divide the interval; I can partition the interval with three grid points 0, 1 and 2; similarly, for x 2 as well. Now, as I said, if we increase the number of grid points, then we will get the better result. Now, increasing number of grid points means we are doing the better approximation. Now, let me consider… Since I am doing manually, let me consider the grid points as this one – k, x 1 k and x 2 k; that means we are considering the grid points. First grid point is 0; second grid point is 1; third grid point is 2. Let me find out the functional values here. What we get? f 11 – that would be is equal to 0; f 1 2… This is f 1 0; this is f 1 1; f 1 1 would be is equal to 1 square minus 4; that is equal to minus 3. Then, functional value at point x equal to x 1 equal to 2; we are having 2 square minus 8; minus 4. Similarly, we will go for functional value say f 2 at these grid points. We will get 0, minus 1; minus x 2 is the point – minus 2. Similarly, go for g 1 k; g 1 0 is equal to 0; g 1 1 is equal to… At point 1, it is equal to 2; and, at point 2, it is 8. Similarly, g 2 k is equal to 0, 8, 32, because this is the g 2 function.

(Refer Slide Time: 50:46)



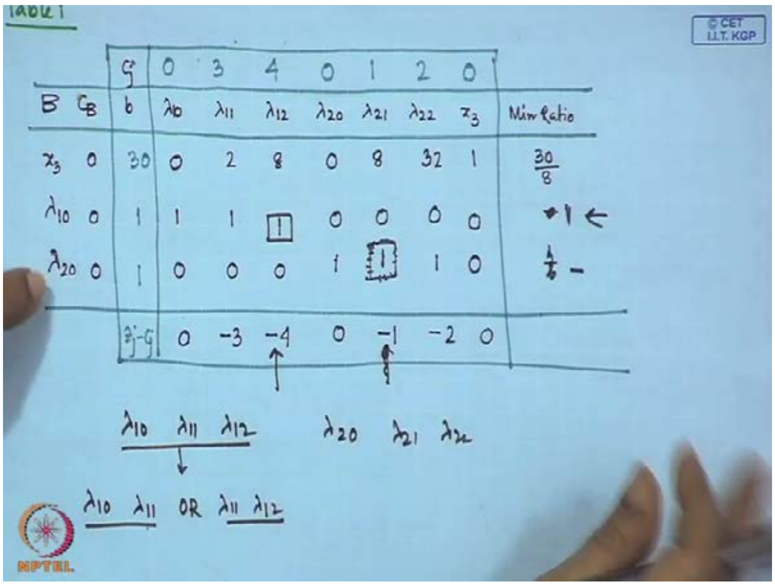

If this is so, we can write down the equivalent linear programming problem in this way. The objective function is minimize; the form was summation is equal to 1 to n; k is equal to 0 to r I; lambda i k f i k; here it is equal to minus 3 lambda 1 1, because lambda 1 0, lambda 1 1, lambda 1 2. We will introduce in this way; lambda 2 0, lambda 2 1, lambda 2 2. Then, it is minus 4 lambda 1 2 minus 1 lambda 2 1 minus 2 lambda 2 2. This is the objective function for us. And, what about the constraint? The constraint would be subject to… This constraint we can write down as the combinations of these – lambda 1 0 into 0, lambda 1 1 into 2, lambda 1 2 into 8 plus lambda 2 1 into 8 and 32 into lambda 2 2 less than is equal 30. And, what is… We are having the condition as… We know the condition is summation lambda i k's would be is equal to 1. Thus, we are having lambda 1 0 plus lambda 1 1 plus lambda 1 2 is equal to 1. Not only that; lambda 2 0, lambda 2 1 plus lambda 2 2 is equal to 1. And, these decision variables are all positive. That is why the non-negativity constraint is satisfied. Just see we are having the minimization of this linear programming problem; thus, we can apply the simplex algorithm very nicely here.

(Refer Slide Time: 53:03)



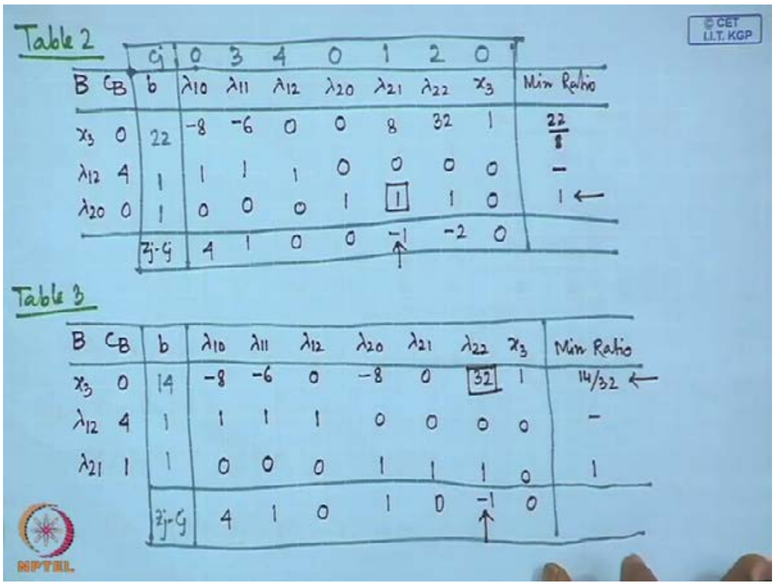

Let me formulate the table for the simplex algorithm. This is the equivalent linear programming problem for the given separable programming problem. As we see that, here we are having six unknowns: lambda 1 0, lambda 1 1 and lambda 1 2. We are having one in equation. And, to make it equation, because we want to apply the simplex algorithm; so, we will introduce the slag variable x 3 and we will get equation to lambda 1 1 plus 8 lambda 1 2 plus 8 lambda 2 1 plus 32 lambda 2 2 plus x 3 is equal to 0.

(Refer Slide Time: 53:40)



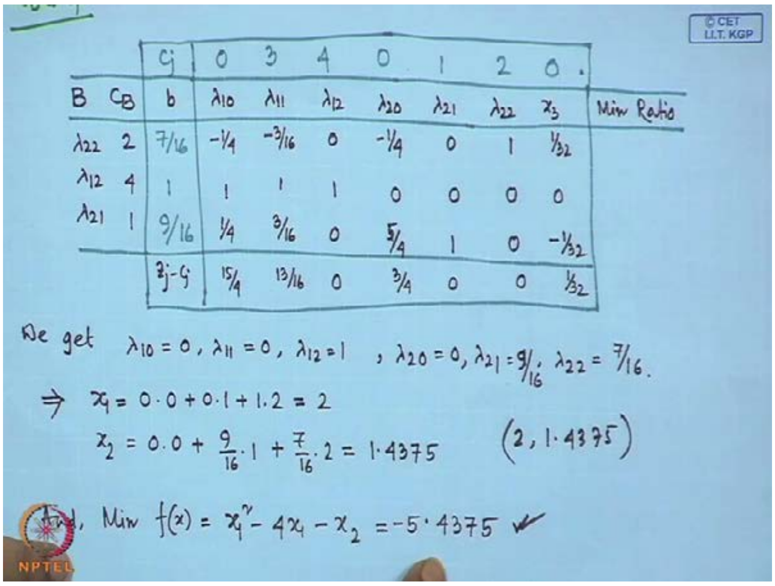
Now, let us just formulate the simplex table in the next. The table 1 – this is the simplex table for us. And, we are calculating the z j minus c j value since this is the minimum value. That is why we are accepting this at that… Thus, lambda 1 2 should be the entering variable in the next table. And, we should see which one is the outgoing variable, because that is more important. As we know the restricted basis entry that, in the set, lambda 1 0, lambda 1 1 and lambda 1 2. There is another set lambda 2 0, lambda 2 1 and lambda 2 2. From here, at the most two lambdas can be in the basis. And, that must be consecutive. That is why this is acceptable; either lambda 1 0, lambda 1 1 would be in the basis or lambda 1 1 and lambda 1 2 must be in the basis. But, as we see, lambda 1 2 is the entering variable; if lambda 1 0 is out going, then only it is acceptable; otherwise, it is not acceptable, because 1 0 and 1 2 cannot stay together. As I have showed, that must be consecutive. Thus, as we see from the minimum ratio, lambda 1 0 is the departing variable. That is why this is quite acceptable to us and this is the pivot element for the next table. That is why in the next basis, we will have x 3, lambda 1 2 and lambda 2 0.

(Refer Slide Time: 54:59)




Again we will just apply the simplex algorithm here; and, we will go for the minimum of z j minus c j. Just look at this z j minus c j value. Here is the minimum minus 2. But, if I consider minus 2 as the minimum; that means lambda 2 2 is the entering variable; lambda 2 0 must be departing; then only lambda 2 2 can enter; otherwise, lambda 2 2 cannot enter, because 2 0 and 2 1, 2 2 – these are not consecutive values. Thus, if we select lambda 2 2 as the entering variable, this must be departing variable. But, this is not acceptable. Thus, we are proceeding to the next minimum, that is, minus 1. And, minus 1 – as we see minus 1, if lambda 2 1 is the entering variable in this case; thus, we can accept it. And, one is the pivot element. We are moving to the next. And, here only one minimum is there. That is why we can select this as the entering variable lambda 2 2.

(Refer Slide Time: 56:06)




And, we will proceed to the next table – table 4. And, the next table gives us the optimal solution like this. And, since here z j minus c j – all are positive; thus, we can accept his as the optimal solution. Then, what is the optimal solution for us? Optimal solution would be lambda 1 0 equal to 0; lambda 1 1 equal to 0. But, lambda 1 2 is 1. But, here lambda 2 0, lambda 2 1 and lambda 2 2 – these are the values. Now, these are the decision variable values for the equivalent linear programming problem. But, we are more interested to get the original separable programming problem. That is why we need to convert this value – lambda values to the x 1, x 2 values. Then only we can say that is the corresponding optimal for the separable programming problem. As we know that, x 1 is equal to lambda 1 0 into the 0 plus lambda 1 1 into 1 plus lambda 1 2 into 2; thus, we get x 1 is equal to 2.

Similarly, we will get x 2 is equal to 2, because x 2 equal to 1.4375; how? Because this is lambda 2 0 into the grid point 0; lambda 2 1 into the grid point – the next grid point 1; and, lambda 2 2 into the next grid point 2. If we just substitute lambda 2 1 and 2 2 value here, we are getting the value for x 2 as 1.4375. Thus, we can declare this is the optimal solution for the original separable programming problem. Again I should mention here this is not very exact solution; this is the approximate solution, because we made the approximation of the nonlinear function with the linear function. And, if I want to have the corresponding optimal value of the objective function, we solve with the maximization problem. Now, we will go back to the minimization problem and we will substitute this x 1 value and x 2 value here and we will get the minimum value of function as minus 5.4375. Thus, we are getting the approximate minimum value for f x.

Now, this is the technique for solving the separable programming problem; separable programming problem is very efficient to solve the separable functions – nonlinear functions. And, this idea can also be extended for the multistage programming problem. And, this is may be the local minima or local maxima depending on the condition of the function. If we go to the convex programming problem, that is, the minimization problem if both the objective function and constraint – both are convex; then, whatever local we are getting; that can be declared as global. That idea holds true for this case as well. That is why if we are interested, we can go for the global maximum depending on the functional nature, that is, the functions involved in the objective function and constraints. And, that is all about today.

Thank you very much.