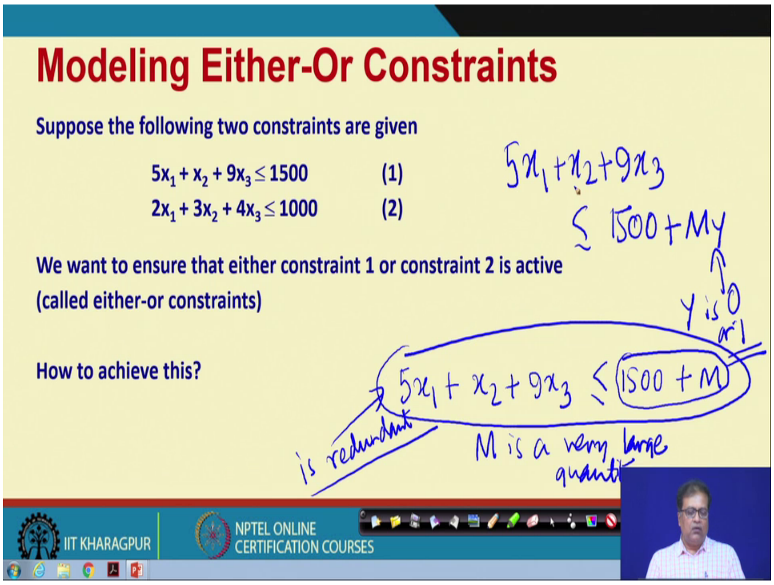
Selected Topics in Decision Modeling
Prof. Biswajit Mahanty
Department of Industrial and Systems Engineering
Indian Institute of Technology, Kharagpur

Lecture - 13
Integer Programming Formulation (Contd.)

Good afternoon to all of you. In our subject Selected Topics in Decision Modeling, today we shall continue our previous discussion on Integer Programming Formulation. In the previous lecture, we have seen some of the formulation examples and let us continue from where we were there in our previous lecture.

So, the first formulation problem that we shall consider is the modeling of either-or constraints. Sometimes it happens like this that we have an integer linear programming problem where the there are series of constraints, but there are some constraints where either a given constraint has to be satisfied or the other constraint has to be satisfied.

(Refer Slide Time: 01:10)



So, let us see what is that problem. So, suppose following two constraints are given that is 5 x 1 plus x 2 plus 9 x 3 less than equal to 1500, and 2 x 1 plus 3 x 2 plus 4 x 3 less than 1000. The idea here is supposing, we have to have only one constraint to be satisfied either the first one or the second one then how should we model that is the question. So, we want to ensure that either constraint 1 or constraint 2 is active called either-or constraints how to achieve this is all right.

So, what should be done. Look here this is where we can cleverly use some binary variables. Now, if we have to have a binary variable, let say y, then how do I connect y. Please remember that these constraints are all less than equal to type is it all right, that means, you know the 1500 that you know left hand side should be less than 1500 on the first occasion, and the left hand side should be less than 1000 in the second occasion.

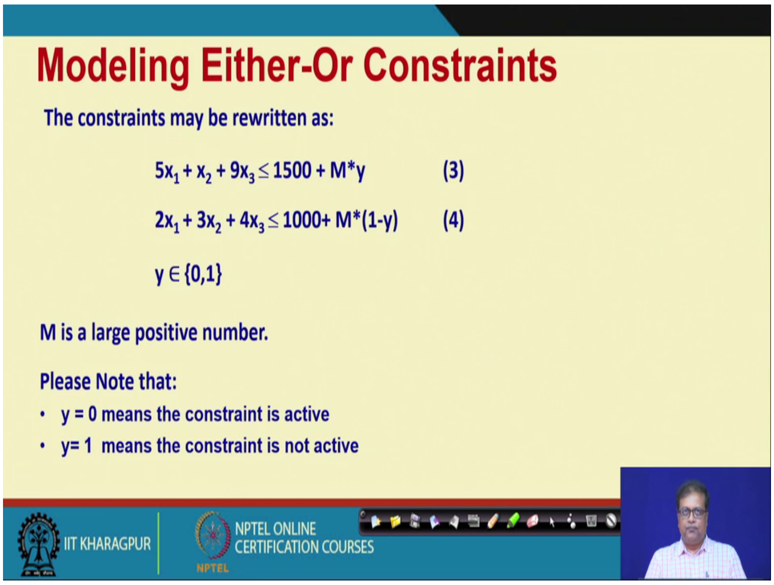
Now, how do you ensure that a particular constraint is not required. So, what to do, you know you add some quantity which is very high. Say let us supposing suppose the first constraint if we can write say 5 x 1 plus x 2 plus 9 x 3 less than equal to 1500 plus M. What is M? M is a very large quantity, so M is a very large quantity. And if we can write like this, then what will happen, then this constraint is redundant. Why? Why this constraint will be redundant, because you know look at the right hand side, the 1500 plus M this is a very high quantity, because M is very large; so M plus 1500 will also be very large.

And you know 5 x 1 x 2 plus 9 x 3 will be always less than a very large quantity. So, what will happen this constraint will be null and void, and therefore not active. Is it all right? The question is now that you know we know now, how to make a particular constraint non active. But then how to cleverly you know really put things in such a manner that you know M is added sometimes; M is not added sometimes, so how to do that? How to ensure, that M is added sometimes and M is not added sometimes.

In fact, if we add M in the first constraint, then we do not add M in the second constraint. Is it all right? Or if we add M in the second constraint; then we do not add M in the first constraint, how to achieve this you know? Make use of the binary variable and then you know right? The equation in such a manner that let us say 5 x 1 plus x 2 plus 9 x 3 less than 1500 plus M y; where y is 0 or 1 that means, y is binary. So, what happens that if y become 0, then the equation is you know active, constraint is active; and if y equal to 1, a constraint is inactive or redundant right. So, that is how to really put.

But then if I put y in 1, and the second one we should not put y, you know we may put what? Think over, we may put 1 minus y. So, if y equal to 0, 1 minus y will be 1; and if 1 minus y equal to 0, then y will be 1 that way we can have either-or constraints. So, that is what we need to do.

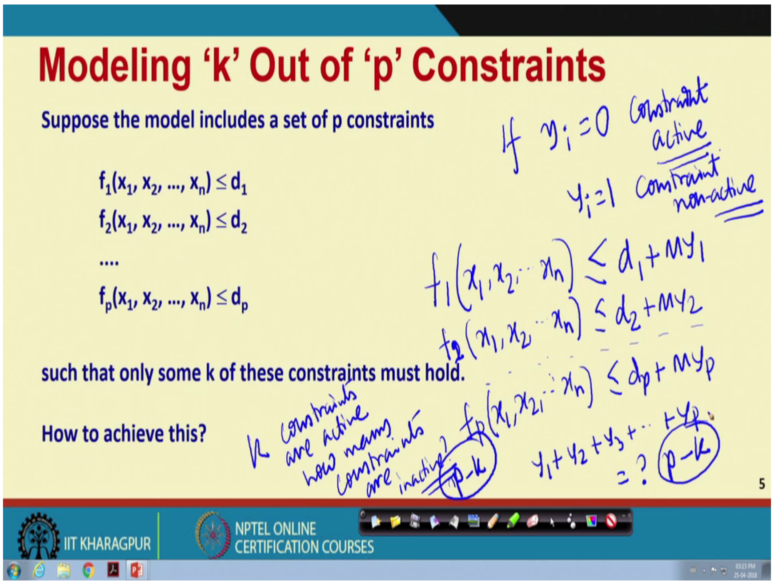
(Refer Slide Time: 06:21)



So, the constraints may be written as 5 x 1 plus x 2 plus 9 x 3 less then equal to 1500 plus M y, 2 x 1 plus 3 x 2 plus 4 x 3 less than equal to 1000 plus M star 1 minus y. And as you have guessed that y is a binary variable. Is it all right? So, that is the way we should model either-or constraints. Now M is definitely a large positive number. Now please note that y equal to 0 means constraint is active; y equal to one means constraint is not active.

That means if y equal to 1, M is added, and 1500 plus M is a very large quantity. And therefore the constraint is you know not going to come in; that means, that LHS will be always less than this quantity at all times right.

(Refer Slide Time: 07:26)



Now, to generalize this we can now have another type of problem which we may call modeling k out of p constraints. So, total number of constraint say a 10, and let us see we say that 7 of them you know should hold; and others should not hold. Then what should we do? So, let us say again the problem is the f 1 x 1 x 2 to x n less than equal to d 1, f 2 you know function of x less than equal to d 2 and dot dot dot up to f p; which is again function of x 1 to x n less than equal to d p.

So, how are we going to ensure that out of these p different constraints, k will hold; remaining k will not hold so how do we go about this? You know only some k of these constraints must hold. How to achieve this? Think over again, we can probably add you know those variables, but should we add y and 1 minus y you know. Just imagine for the time being in the previous problem, suppose we have you know added y and 1 minus y instead if we added y and z, so what would have happened? We require an additional constraint that y plus z equal to 1.

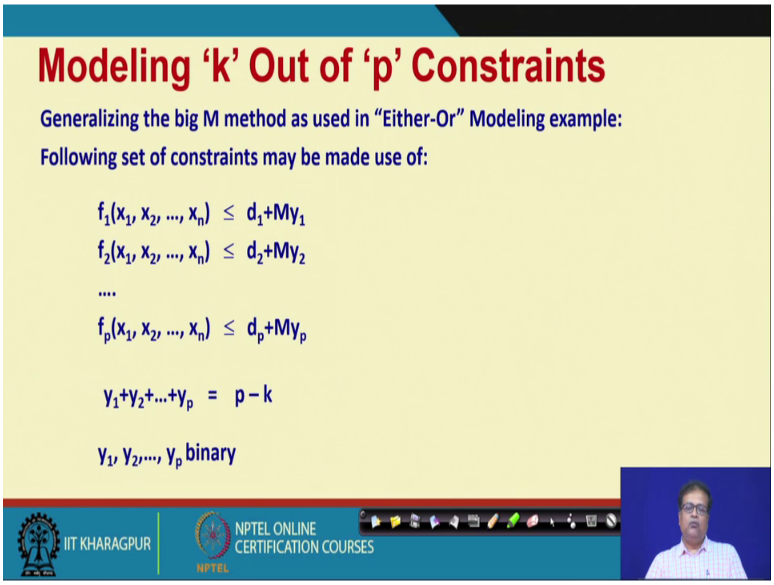
But you know in this case if we really think, that all these constraints say the first constraints. Supposing you know we say that f 1 x 1 x 2 dot dot dot to x n less than equal to d 1 plus M y 1 all right? So, this is how we can implement by going the similar logic; And in the second case supposing, we put less than equal to d 2 plus M y 2; And the last one will be f p, the second one will be f 2 f p x 1 x 2 to x n less than equal to d p plus M y p so all these constraints. So, what can you infer or what is your idea about, y 1 plus y

2 plus y 3 plus y p should be equal to how much? Should it be k or something else just think over. See the point is if y i equal to 0 then constraint is active or non active just think; if y i equal to 0, then constraint is active, so we want the number of active constraints to be k.

And y i equal to 1 that means, the constraint is redundant or non active. So, if the constraint so, you see p constraints, k constraints are active. So, how many constraints are inactive? How many constraints are inactive? Just think over. You know if that means, an inactive constraints will be added to 1. Is it all right? k constraints are active that that means, k of them are 0, so how many are non-zero that means, inactive constraints it should be p minus k.

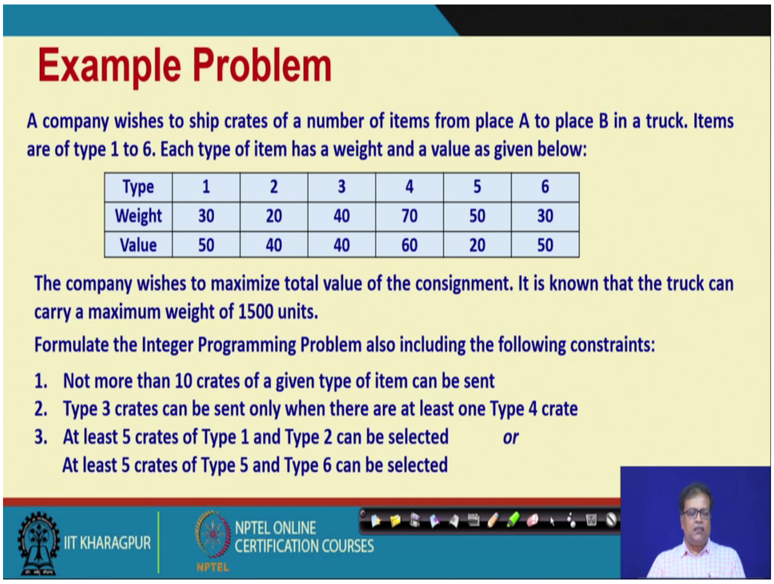
So, it is the p minus k which should be the sum that is the all the y i's when you add this sum should be p minus k. Is it all right? Because that is the number of inactive constraints and those values should be one, remaining value should be zero; that means, k constraints having y i value equal to 0 and therefore, those constraints will be active. I hope you got the essential idea.

(Refer Slide Time: 13:07)



So, let us see how it is that you know modeling k out of p constraints, the generalizing the big M method as used in either-or modeling example following set of constraints that is the first one, the function of f less than d 1 plus M y 1; second one less than equal to d 2 plus M y 2; third one like this, the last one d p plus M y p. And all this sum of y 1 y 2 up to y p should be p minus k as we have explained. And also to note all this y i's are binary right. So, this is how we model either-or constraints or in a generalized way let us say the k out of p constraints.

(Refer Slide Time: 14:04)

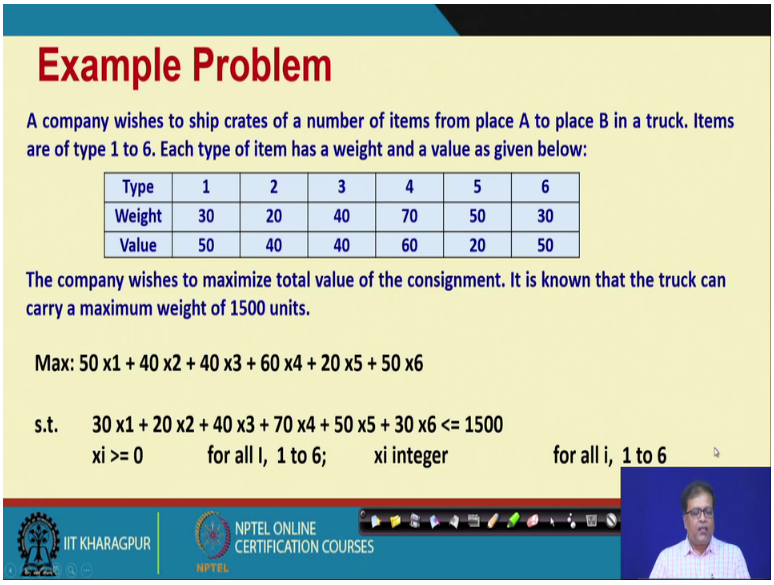


So, once we know this let us take an example problem. A company wishes to ship crates of a number of items from place A to place B in a truck. There are 6 types of items, for each item has a weight and has a value, it is like a kind of knapsack problem. The company wishes to maximize total value of consignment, it is known that the truck can carry a maximum weight of 1500 units.

Formulate the integer programming problem also including the following constraints number 1, not more than 10 crates of a given type of item can be sent right. Type 3 crates can be sent only when there are at least one type 4 crate. At least 5 crates of type 1 and type 2 can be selected or at least 5 crates of type 5 and type 6 can be selected. Is it all right?

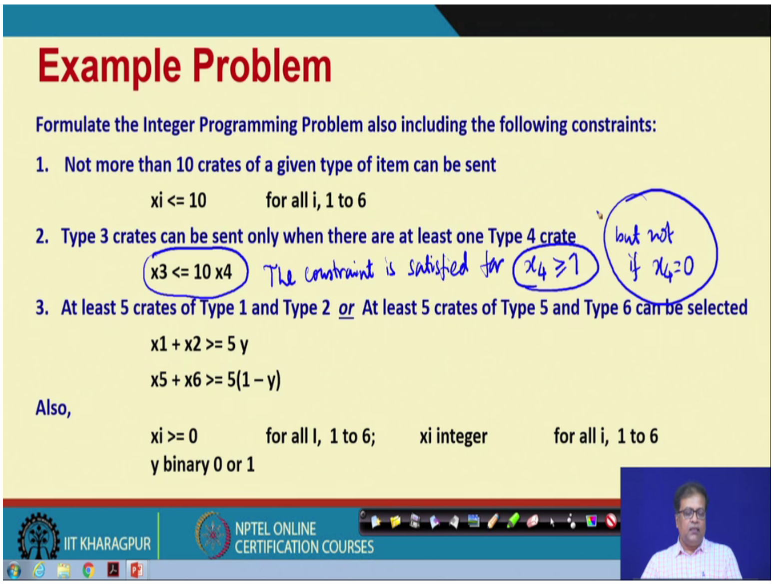
So, how do we go about this? So, first part of the problem is very simple that is it is a standard knapsack problem, so we can just formulate this in the form of a knapsack problem formulation.

(Refer Slide Time: 15:34)



That is maximize the total value which is 50 x 1 40 x 2 etcetera, subject to the constraint that you know this should be within that 1500 limit that means, the total value you know sorry total weight should be within the 1500 units right. So, 30 x 1 etcetera should be less than equal to 1500, and x i should be greater than equal to 0. We have already seen how to model such problems, in the form of a integer programming problem. So, this part is simple.

(Refer Slide Time: 16:17)



But the additional constraints, let us see how we have modeled. The first constraint that not more than 10 crates of a given item can be sent, this is very straight forward; that is that for every item x i should be less than equal to 10 right. So, it is very simple that that means, x 1 should be less than equal to 10, x 2 should be less than equal to 10 and like that. The second one look at very carefully type 3 crates can be sent only when there are at least one type 4 crates. Is it all right?

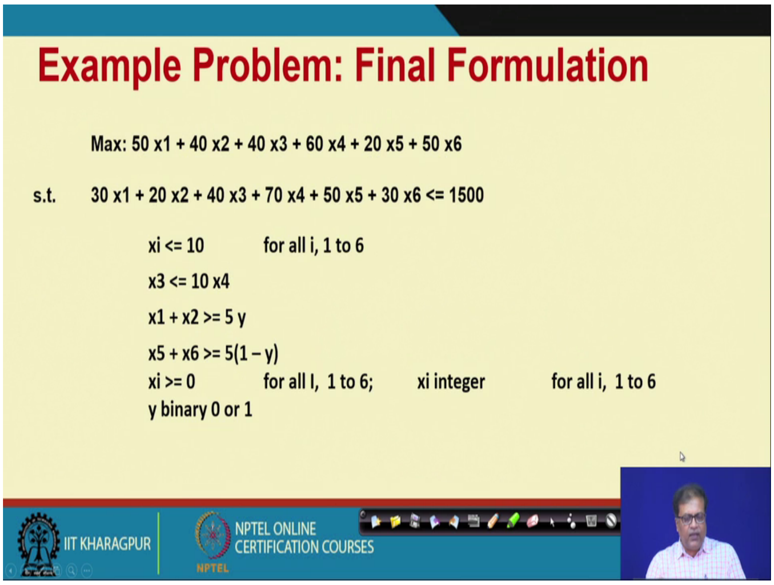
So, it means that if there is a type 3 crate even one then there should be at least one type 4 crate. See what should be the maximum value of x 3 it is 10, so look here x 4 you know can be, you know x 4 should be if x 4 is 0; then 10 x 4 will be 0 also, but then you know that will not satisfy this particular constraint, just imagine for the time being that x 3 equal to 2 and x 4 equal to 0, then this constraint will be violated.

But, if x 3 equal to any value within up to 10, and x 4 equal to 1 then this or any number more than 1 you know this constraint will be satisfied. So, we can write here the constraint is satisfied for x 4 greater than equal to 1 right; but not if x 4 equal to 0 ok. So, you know automatically you can understand that there should be at least one type 4 crate this constraint will be satisfied. A look at the third one at least 5 crates of type 1, and type 2; or at least 5 crates of type 5 and type 6 can be selected. So, initially let us think of you know the let us take out this part first, suppose if they do not exist; suppose that part does not exist.

So, what would be then that at least 5 crates of type 1 and type 2 that means, x 1 plus x 2 should be greater than equal to 5. And at least 5 crates of type 5 and type 6, it is x 5 plus x 6 greater than equal to 5 right, so these constraints are satisfied. But then you know if you satisfy the first one, then it is not required for the second one; or if you satisfy the second one, it is not required for the first one. So, that is where we can bring in our concept of binary variables that is you know we multiply it with a binary variable y. If y equal to 0, then you know the constraint will be always satisfied is all right? That means, you do not really need at least 5.

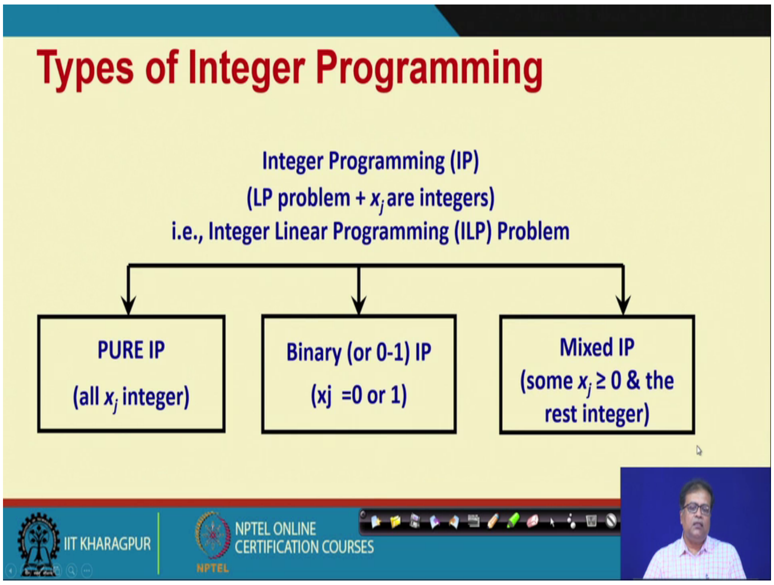
But then if y equal to 0, 1 minus y will be 1; then in the second case must be satisfied, and if y equal to 1, the 1 minus y will be 0; so the first case should definitely be satisfied. It is all right? So, this is the interesting way by which you can model such kind of constraints. Obviously, the y is binary and x i is a value from 1 to 6. So, you have seen how these kind of problems are solved or not really solved, formulated the formulation part of it you know you can understand. The solution we shall discuss a little later.

(Refer Slide Time: 20:43)



Now, let us see now, that is the final formulation.

(Refer Slide Time: 20:47)

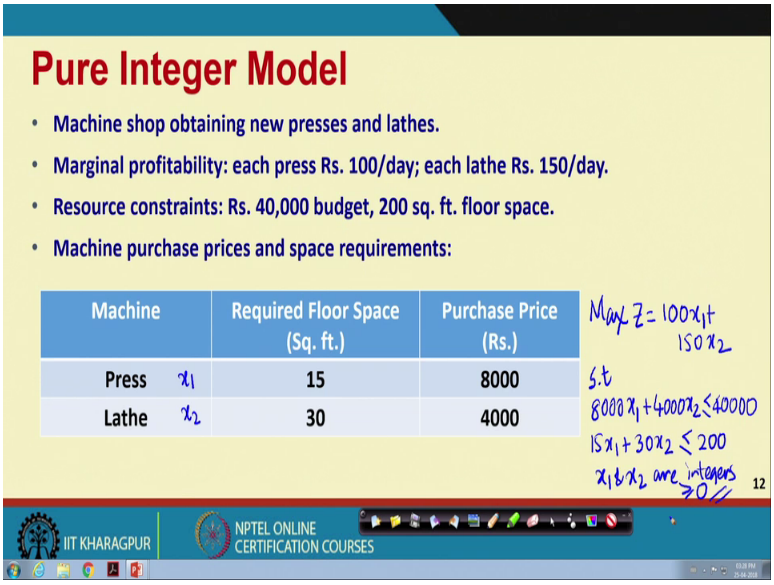


So, let us see; what are the different types of integer programming problems that we can have. The integer programming problem, mostly we shall discuss integer linear programming problem. I mean you should know that the integer programming problem includes, obviously non-integer programming problems also. But then for the time being we shall not really discuss about non integer, non-linear kind of programming problems, we shall mainly discuss the integer linear programming problem that is ILP.

Now, you know such problems are essentially of three types; one is pure integer programming problem, binary integer programming problem and mixed integer programming problem. The pure integer programming problems, where all the x j values are integers; The binary problems are such where the value could be 0 or 1. Is it all right? And the mixed integer, some of these variables are regular they need not be integers whereas, the others should be integers right.

So, as you can see all of these are special class of problems, because while we have very powerful simplex algorithms available for solving linear programming problems. But movement we have integer programming considerations whether pure or binary, or mixed solution becomes difficult, because we just cannot use the simplex programming solutions and the way we know right, we may have to either modify or use some other methods.

(Refer Slide Time: 22:47)

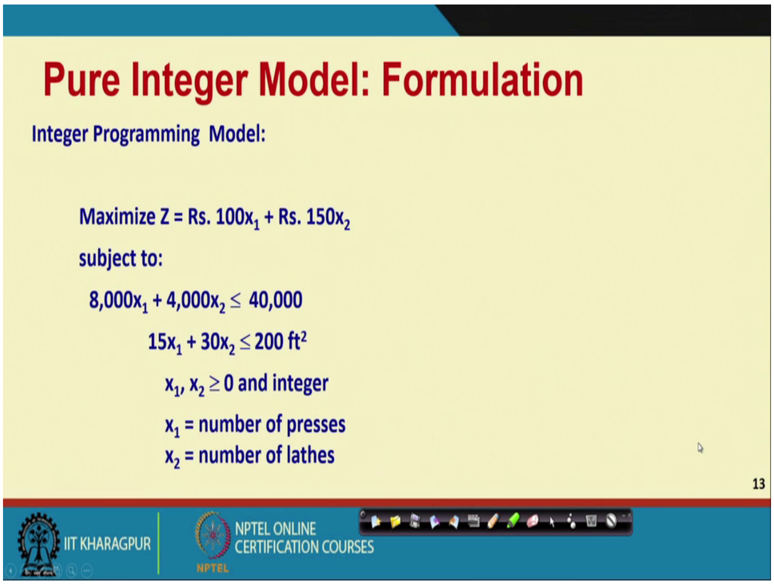


So, let see some examples. Let see an example of a pure integer model, so let say a machine shop is planning to purchase a new press and a new lathe, is all right. So, the margin and profitability each press rupees 100 per day; and each lathe is rupees 150 per day. We want to buy number of presses and number of lathe, but then our total budget is 40000 rupees only, and we have 200 square feet floor space. And these are the machine details, the price and the required floor space details is it all right. So, how do we model this right, so what should we do? We should take then two variables, let us call them x 1 and x 2. So, let say press v by x 1 number and lathe x 2 number. Is it all right?

So, what is the objective function? The objective function is to maximize profitability, so what it should be maximize? Say Z equal to 100 x 1 plus 150 x 2, so this is the objective function. What are some constraints? The constraints will be subject to first of all the budget constraints, so 8000 x 1 plus 4000 x 2 should be less than 4000 that is the budget constraint. And the required floor space constraints, 15 x 1 plus 30 x 2 less than equal to 200; and x 1 and x 2 are integers. Obviously, they are greater than equal to 0, but they are also integers.

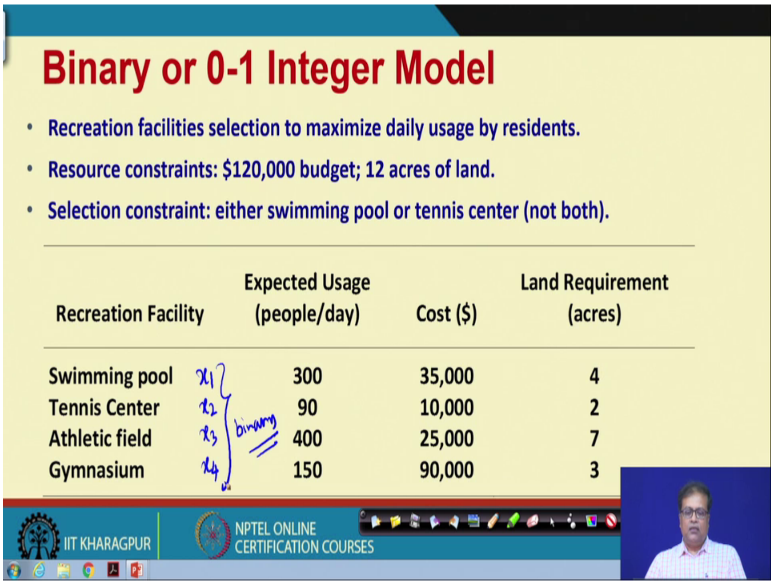
Because, you cannot buy some point 5 press or point 5 lathe and all that, Is it all right. So, that is what is essentially are pure integer models right.

(Refer Slide Time: 25:32)



So, let us see then you know the formulation once again, maximize Z equal to rupees 100 x 1 plus 150 x 2 subject to 8000 x 1 plus 4000 x 2 less than equal to 40000; 15 x 1 plus 30 x 2 less than equal to 200 I mean within the floor space; and x 1 and x 2 greater than equal to 0 and integer right, so this is a pure integer model.

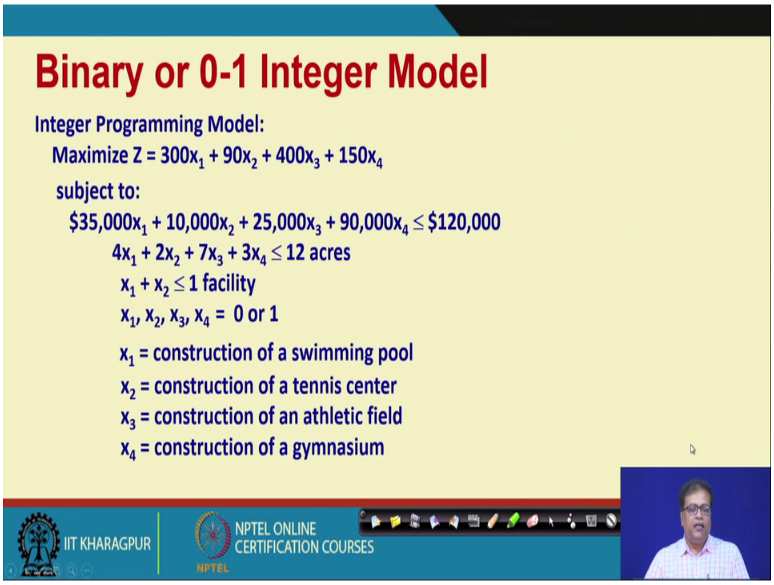
(Refer Slide Time: 26:06)



The next one a binary model; so, let us say recreation facilities are to be created and we have a budget, and we have the total amount of land. And there are four types of recreation facilities are to be created swimming pool, tennis center, athletic field and gymnasium right; each has a cost and expected usage, and land requirements.

So, here the difference is that you know unlike the lathe and press, we were not going to really create more than one swimming pool or one tennis center, one athletic field or one gymnasium is all right. That means, either we create a swimming pool or we do not create, it is all right. So, if that is so then again assuming 4 variables, x 1 x 2 x 3 x 4 they all will be binary; and they will be subject to the usual objective function, and the constraints.

(Refer Slide Time: 27:22)



So, what are they let us see them the thing will be the maximize Z equal to 300 x 1 plus 90 x 2 plus 400 x 3 plus 150 x 4 why because these are our expected usage, we want to maximize them; subject to the monitory constraint and the land constraint, is it all right.

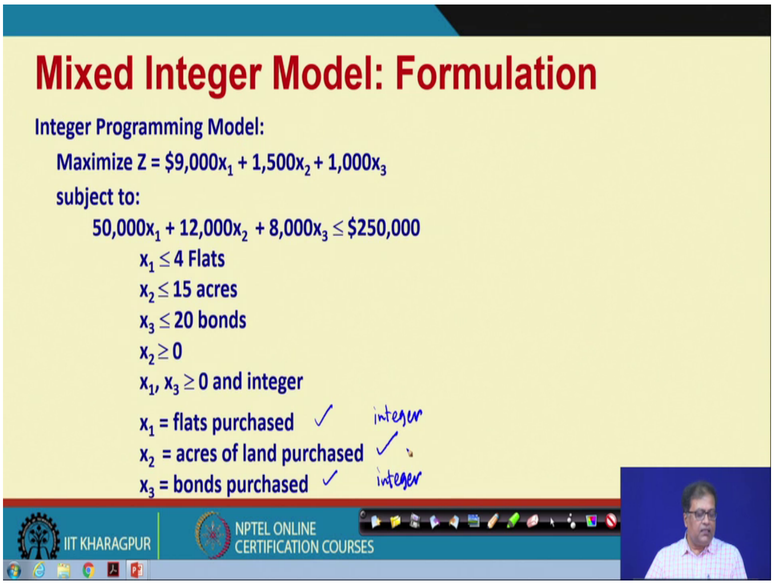
And you know x 1 plus x 2 less than equal to 1 that is an additional constraint that we have given that is that you know we should create either swimming pool or tennis center not both so right. So, it is also said that you cannot have both swimming pool and tennis center. So, if that is so then we have x 1 plus x 2 less than equal to 1; and all these x 1, x 2, x 3, x 4 are 0 or 1, is it all right. So, I hope you understood about the binary integer modeling problems.

(Refer Slide Time: 28:23)

The next one is the mixed integer models. Now, suppose I have certain money available for investment providing return for you know flat, land and municipal bonds, is it all right. So, the flat cost is given, and if you sell after one year, you are going to get a profit of 9000. The land also has a cost per acre, and you will make a profit of 1500 if sold after one year; I mean per piece of land.

The municipal bond on the other hand costs 8000 per bond and you have 1000 profit if sold after one year. So, only 4 flats, 15 acres of land, and 20 municipal bonds are available. So, how do we go about it?

(Refer Slide Time: 29:24)



So, again we assume x 1 x 2 x 3 flat purchased right, then acres of land and bonds. But please remember that while the flat has to be an integer, the bond has to be an integer, there is no such need about the land. You can purchase you know land need not be integer, you can buy let say 2 point 5 acres land there is no problem, it is all right.

So, this is that is how it is a mixed integer problem. And obviously, if you if you look at the previous slide, then the you know the returns are 8000, 1000, for 9000, 1500 and 1000 respectively.

So, 9000 x 1 plus 1500 x 2 plus 1000 x 3 that is an objective function to be maximized subject to you know the 50000, 12000 and 8000 which are the costs, so 50000 x 1 12000 x 2 plus 8000 x 3 should be less than the total amount available; x 1 is less than 4 flats, x 2 is less than equal to 15 acres and x 3 is equal to 20 bonds. And they are all greater than equal to 0, but additionally x 1 and x 3 are integers right.

So, we have seen that how different kinds of integer programming problems are formulated. In our next lecture, we shall see how we can solve such integer programming problems.