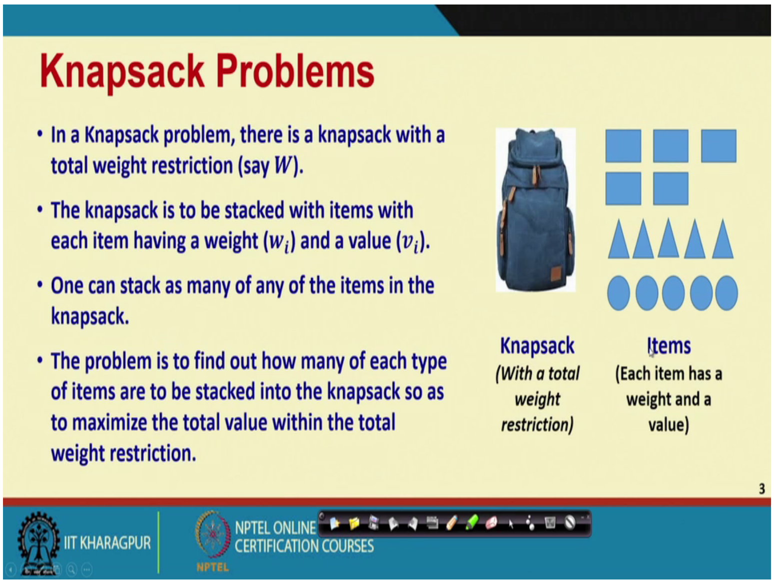
Selected Topics in Decision Modeling
Prof. Biswajit Mahanty
Department of Industrial and Systems Engineering
Indian Institute of Technology, Kharagpur

Lecture - 07
Knapsack Problem

Good morning to all of you, in our course Selected Topics in Decision Modeling. Today let us have the lecture number 7 that is the Knapsack Problem. So, the knapsack problem as you all know is you know about in a assuming there is a knapsack which you know this one.

(Refer Slide Time: 00:34)



Suppose it has got a total weight restriction and we have a number of different items and each item has a weight and a value.

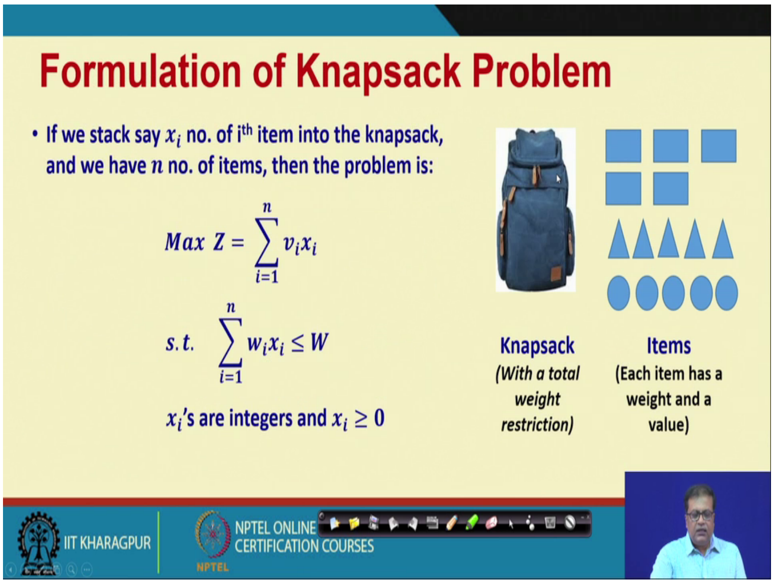
So, what happens the idea is that how do I fill up this particular knapsack with you know number of items; when we say a number of items it does not really mean that you know I just have one of each. It really means I can have a number of first one and number of second one a number of third one or we may not have any from a particular group. How do we do that, so that we can carry the maximum possible value.

So, the idea here is to fill up the knapsack with the maximum value within the weight restriction.

And how many of each type of items therefore, that to be stacked into the knapsack. So, as to maximize the total value within the restriction now can you think of some problem which you know really resembles a knapsack problem. If you really recall then we have done one such problem that is called the cargo loading problem.

So, you see the cargo loading problem what is really happening that there was a truck the truck has to be filled with certain number of items and within the total weight restriction of the truck and the idea was that how do I carry items of the maximum possible value. So, that itself in a generalized form can be called a Knapsack problem.

(Refer Slide Time: 02:46)



Now, let us see that how do I formulate a knapsack problem. So, let us say we will stack xi number of i-th item into the knapsack and we have n number of items. So, total number of items are n and I is a given item say maybe the triangle here. So, let us say i is a triangle one is the rectangle and two is the triangle and three is these circles.

Like these suppose there more items than 4 5 etcetera the i-th one let us say one of them and therefore, the maximization will be maximize Z equal to sum over i to n vi xi. Is it all right? So, maximizing the total value and subject to a restriction which is basically w i x i; that means, the total weight of all these items should be less than equal to W that is the total weight of the knapsack and xi's this is the most important thing the individual number of items should be integers n right, this is the restriction which is the most important here they are all integers.

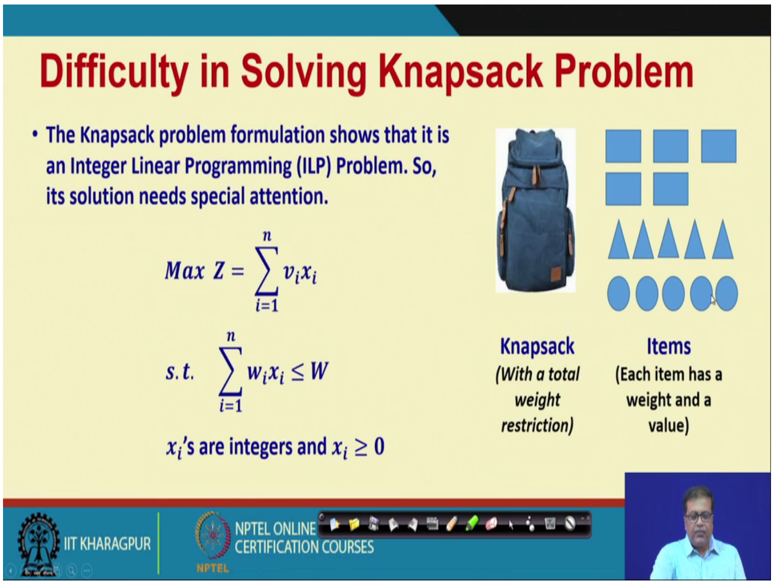
And xi should be greater than equal to 0. Having said that let me also tell you that this is not the only type of knapsack problem, there are knapsack problems which may be called as 0 1 knapsack problem where the item could be there or may not be there. So, those will be a special class of problem. There are also partial knapsack problems; in partial knapsack problem what really happens you know you can have fractional amount of items as well right.

So, assuming that is really possible suppose the items are really quantifiable and you can actually you know take fractional amount in certain cases. Maybe in case of groceries you make some new packets out of some stacks, suppose you can really do that then you may also have a partial knapsack problem.

But forgetting about all those variations the kind of problem that we are going to have in this particular you know lecture is only the kind of knapsack problem that I have described that is xi's are integers their values are 0 to certain numbers and also again you know another variation that the number of items could be limited.

So, suppose we do have here only 5 number of rectangles, triangles and circles, in another problem we may have maybe less number of rectangles so; obviously, we cannot take more than them. So, we have not put that restriction. So, let us assume that there is really no limit to the one number of items of a given variety right. So, that is how is the knapsack problem, that we have considered here.

(Refer Slide Time: 06:14)

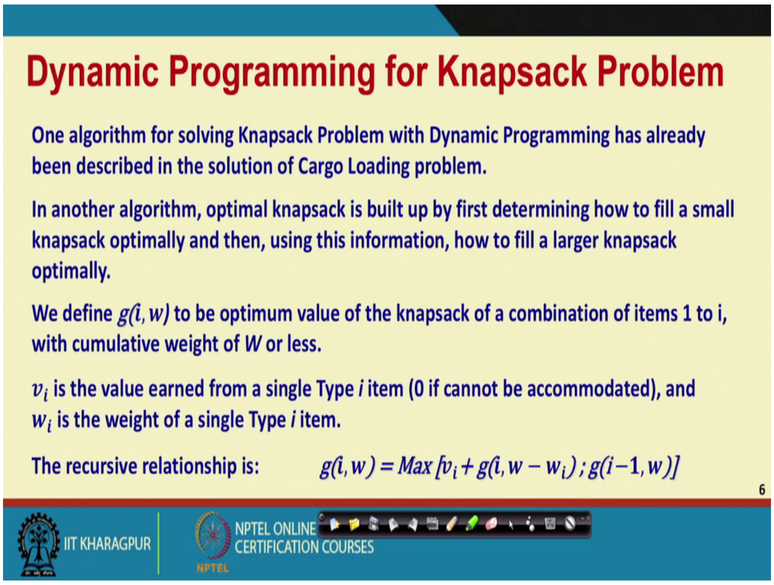


So, having said that what is the difficulty; in solving such knapsack problems; You see this problem really falls into the class of integer linear programming. So, you see integer linear programming problems they which we shall take up in a big way in our next set of lectures.

One thing to not here is that integers problems cannot be solved simply by the methods such as a linear programming. You know the Simplex method if you use as it is without integer considerations you cannot solve the integer linear programming problems.

So, since this is an integer linear programming method we need some other method to solve such problems that is the importance of the knapsack problems.

(Refer Slide Time: 07:13)



So, what is that method that are available you know we shall discuss here as you all know it is nothing but the dynamic programming method. So, there it does not mean that dynamic programming is the only method of solving knapsack problems.

There are other methods as well, but within the broad context of dynamic programming we discuss here only the dynamic programming methods there are two broad methods right or algorithms the first one is the same as we have used in the solution of the cargo loading problem is it all right. So, you probably remember that how exactly we have gone ahead and solved the cargo loading problem.

But you know there another algorithm which is also very popular and that algorithm we shall discuss in this particular lecture. So, what happens in this particular algorithm an optimal knapsack is built up by first determining how to fill a small knapsack optimally and then using this information how to fill a larger knapsack optimally.

So, like what we really do we take up a small knapsack suppose our original weight restriction is let us say 10 kg or 10 tons then first we take a very small knapsack, let us say 1 ton is it all right and we take only 1 item right. So, we take only one item and let us say we take only 1 ton and see what kind of results are you going to get.

Then we can you know increase it and lest say call it say 2 tons then 3 tons 4 tons. So, like this we increase the knapsack size and; however, calculate the optimal number of item one in each such case. Once this is done then we take a bigger problem that what is the optimal solution for if I take not one item, but item 1 and item 2 together and why we do so? We make use of the optimal solution that we have found you know combining item 1 and item 2.

So, once again we will start from let us say 1 ton, 2 ton, 3 ton and go all the way up to 10 tons. So, what really happened that we take a smaller knapsack and slowly increase the size and in the first stage we take only one item and see for each such knapsack steps what is the optimal and how exactly we can you know fill up the knapsack what is the optimal result.

So, the optimal result that we get for let us say one item can be used for you know at the subsequent stage where we have items 1 and 2. So, again by similar procedure if I find the optimal solutions for two items using them we can actually you know solve the for the items 1, 2 and 3 like these we keep on increasing the items and finally, when we get the final solution by utilizing the previous stage results the final solution is available.

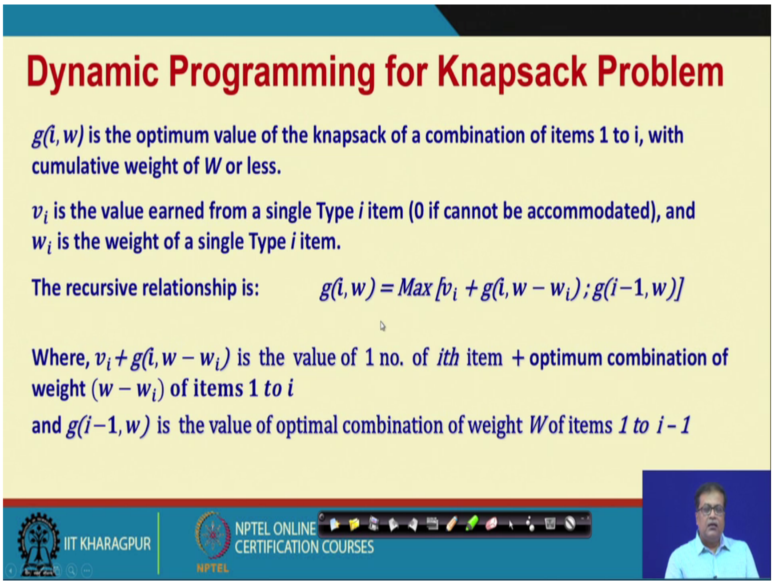
So, in mathematical language we defining gi w to be optimum value of the knapsack of the combination of items 1 to i with cumulated weight or cumulative weight of w or less right.

So, exactly in a in between stage let us say w are in the i-th stage. So, an i-th stage we take a combination of items one to i and weight should be considered from the minimum to w which is the weight restricted value the final you know weight that we should have and let us call it g iw.

So, vi is a value and from a single type of i item and see supposing vi is a value of an single item i this all right, but suppose it cannot be accommodated suppose our knapsack size at a given point is only 2 tons and the item given item is weight is 3 tons.

And; obviously, cannot accommodate in that case the value will be 0 right and let us say w i is the weight of a single i type i item when the recursive relationship look at it very carefully g i w equal to maximize v i plus g i comma w minus w i gi minus 1 w. So, what is this recursive relationship and what does it mean.

(Refer Slide Time: 12:30)



So, let us see our next slide and try to understand that. So, as I said before that g iw is optimum value of the knapsack of a combination of items 1 to i with cumulative weight of w or less. So, vi is the value and w i is the weight and this is our recursive relationship. So, first of all vi is the value of an item i gi w minus w i what is wi; wi is the weight corresponding to that item i. So, if I deduct that item weight then w minus wi is the weight that I have left. So, what is the optimal value for that?

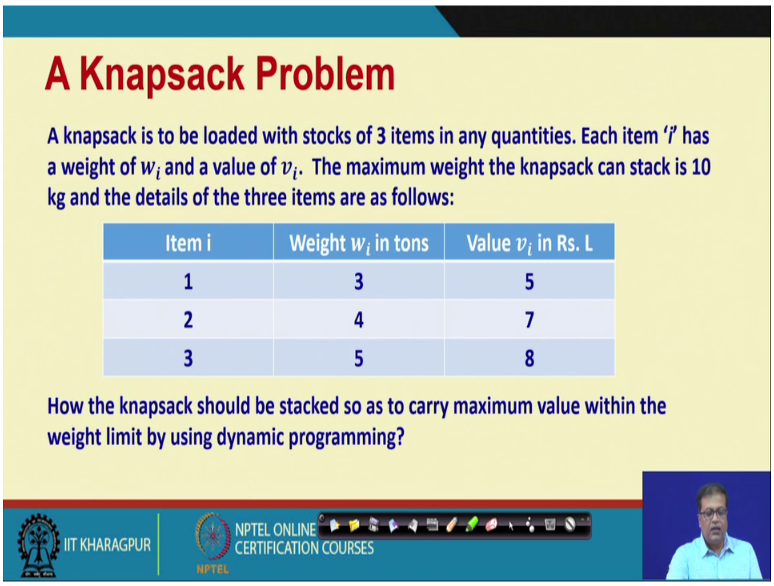
So, use the same function please remember that while i fill up g i w i start from the minimum weight and move up to maximum weight w. So, already I have computer g i w minus w i because it is less than is g i w is it all right. So, let us say that total weight is ten tons and these i 1 single item of item given item is a 4 tons and then g i 6 I have already computed. So, suppose i equal to 2 w equal to 6 then you know, let us write it here that g you know 2 6 g 2 comma 6.

What will be g? Sorry total is 10. So, let us use the pen here. So, you see that g g 2 comma 10 right g 2 comma 10 will be what this g 2 comma 10 will be the weight of the value with the second item and the value of the second item suppose that value is 7 then this should be g 2 comma 6. Why g 2 comma 6 because you know 2 is the second items 6 is the remaining weight that I have available. So, this is the first part of the thing right.

So, having these kind of a value that we have the recursive relationship that g i w will be max vi plus g i w minus wi; however, at the previous stage that is i plus 1. So, if i equal to 2 then in the first stage where we consider only one item what was the optimal value. So, that we have had. So, this value or this value whatever is the highest we take the maximum of these two and that will be our g i w right. So, this is a recursive relationship that we are going to use.

So, in mathematical language it is the value of one number of i-th item plus optimum combination of weight w minus w i of items 1 to i in the same set of iterations and g i minus 1 w is th value of optimal combination of weight w of items of 1 to i minus 1. Now, let us go further and see how do I do this in the in the given example right. So, let us say a given knapsack problem right.

(Refer Slide Time: 16:20)

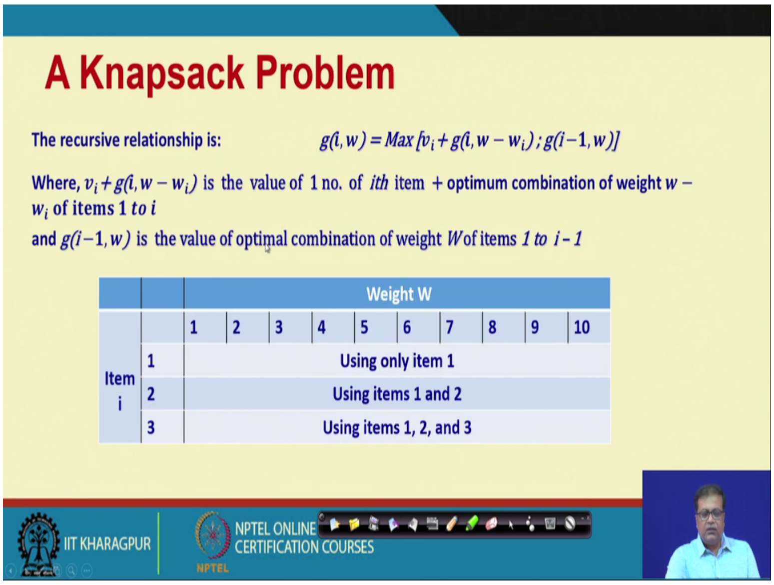


So, let us take a knapsack problem; the knapsack problem is the knapsack is to be loaded with the stocks of 3 items in any quantities. Each item has a weight w i, I mean item i and the value vi. The maximum weight of the knapsack is 10 kg and these three items have got weight 3 tons, value 5 lakhs let us say in rupees lakhs; I call it rupees L, second item 4 tons and 7 lakhs, third item 5 tons and 8 lakhs.

So, what is the optimum combination within the knapsack of 10 kg? I mean; obviously, if I say tons then it cannot be in kg it should be 10 tons.

So, 10 tons how do I you know maximize the value that is the knapsack problem. So, how do I go ahead with this particular problem?

(Refer Slide Time: 17:26)

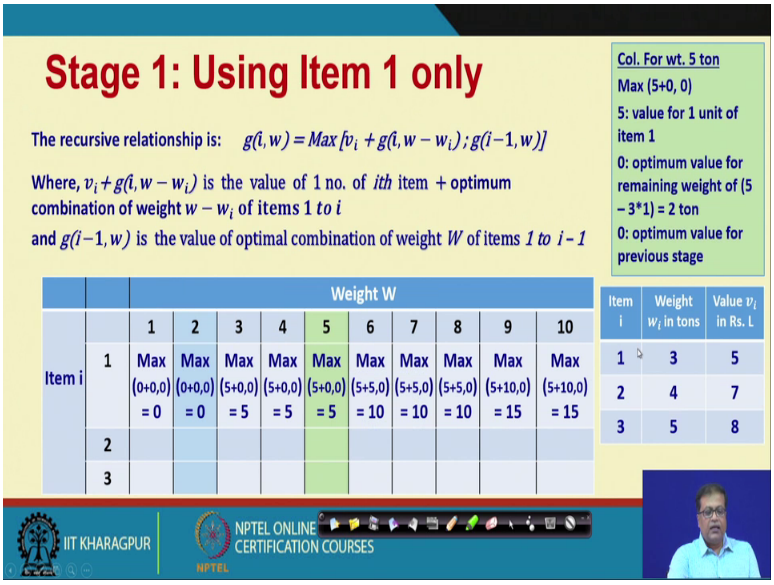


So, look here that recursive relationship that I said that will be used. So, it is written here for your reference what we do really it is the knapsack size could be anything from 1 to 2.

So, I start taking smaller knapsacks first starting with only 1 item and fill up for the smallest knapsack that is knapsack of 1 weight 1 ton, knapsack of 2 ton knapsack of 3 tons 4 tons 5 tons 6 up to 10 tons right and then at the second stage we use items 1 and 2 both and in the final stage we use items 1, 2 and 3 all together is all right. So, these are the three stages.

Now, what could be the state the stage is clear that there are three stages in the first stage. We consider only item 1 in the second stage we consider both items 1 and 2 and in the third stage we really do the items 1 2 and 3 and you know the state could be measured by the amount of weight that we are going to allocate.

(Refer Slide Time: 18:43)



So, this is a recursive relationship and this is how we go ahead with the problem. So, let us see at the stage 1 if I have only 1 item that is the first item only how do I use the formula. See at the very first you know at the first if I have only a 1 ton you know the simplest problem and 1 ton Knapsack and only 1 item right. So, we have only one item and we have a single Knapsack of one.

So, what will happen? The maxima is 0. What is 0? Zero is the value. Why the value is 0 because the first item weight is 3 tons and its value is 5 lakhs, but we do not have 3 tons we have only 1 ton. So, we cannot really put an item 1 here so; obviously, the value cannot come in so, it should be 0. At the next stage you know the other 0 is basically what is the item, suppose it cannot be accommodated. So, g i w minus w i will be 0 also because you see there is 0 1 ton available.

So, we could not accommodate. So, there is no question of w i. So, therefore, this term will not exist right. So, that will be 0 also and there is no previous you know there is no previous item. So, since it is the first item so; obviously, the optimal value for the previous item will be 0 also.

So, it will be 0. So, like this we will compute. So, as an example let us see what happens if the weight is 5 tons. If the weight is 5 tons we what we did you see now it will be very clear what is the weight of a single what is the value of a single item 1 right. So, we have taken item 1. So, we are here. So, this is item 1. So, a single item 1 the value is 5 right. So, this value of 5 that is the vi. So, look here this vi we will talk about.

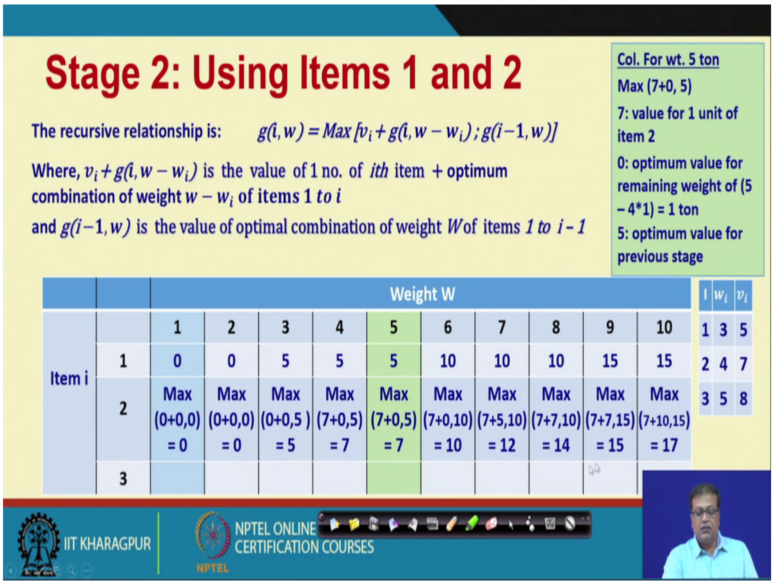
So, what is v i? The vi is this 5 is it all right. So, this 5 is put here. So, this 5 and what is this 0? The 0 could be found out look here the weight here is 3. So, what is the weight? Weight is 3. So, what we do this 5 that we have an knapsack of 5.

So, 5 minus 3; how much is 5 minus 3? So, w minus w i will become 5 minus 3 equal to 2 right. So, 2 tons is what we left. So, what will be the 2 tons? So, you see what is the g item 2 with 2 tons; that means, g 12; this is the g 12 and what is that value? It is 0. So, that 0 is taken here, I hope you understand and there is no previous item. So, it will be 0.

So, what is the net? Net value should be 5. So, this is all explained here. So, column for weight 5 ton maximize 5 plus 0; 5 is the value of one unit of item 1, 0 is the optimal value for remaining weight of 5 minus 3 in to 1 that is 2 ton and 0 is the optimal value for the previous stage right. So, this is how you can compute let us say that a single item.

Now, if we have to go one stage further if let us say we have to go next stage you know supposing another calculation let us see. So, here something is slightly different what happens similar, calculation what happens if we have I know knapsack size of 10 tons. So, in the 10 tons maximize 5 plus 10 comma 0 where 5 is the value of 1 unit of item 1 as you have seen earlier.

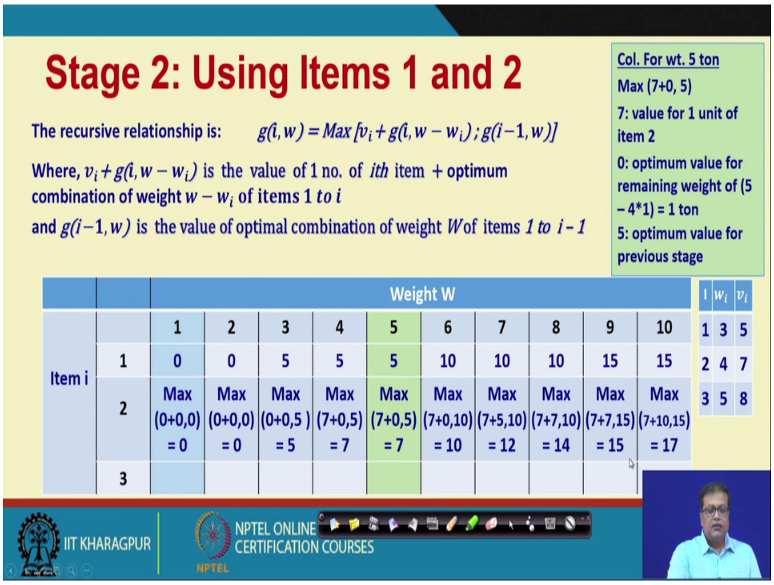
(Refer Slide Time: 23:14)



So, you see this is the same value that is 5 right. So, one sec we are in the previous slide yeah. So, this is the 5 that is the value the 10 is essentially what is to be 10 ten will be look here 3 is the weight of a single item. So, 10 minus 3 7 ton knapsack what is the optimal solution. So, 7 ton 7 ton knapsack that is the g i w minus w i the optimal solution was 10.

So, these 10 and these 5 will be 15 and the previous is 0 so; obviously, this value should be 15. So, now, I hope you understood that how exactly we calculate all the knapsack values if we and the stage 1 and if we take let us say only 1 item. Go to the next stage ok.

(Refer Slide Time: 24:16)



So, as I go to the next stage then we take 2 items that is item 1 and item 2 together. So, when we take item 1 and item 2 together; now you look at the recursive relationship once again. Now, you tell me what is this g i minus 1 w what is this term? This term will come into play. This term is nothing, but the optimal value for a given size knapsack suppose knapsack size is w the optimal value at item 1 level.

So, the highlighted one let us see the highlighted one is a 5 ton Knapsack. So, if I take a 5 ton knapsack what will be your g i minus 1 5? We in currently we are in the second item right. So, we are now considering item 2 that is the decision the state is the stage is the using both items 1 and 2 please understand.

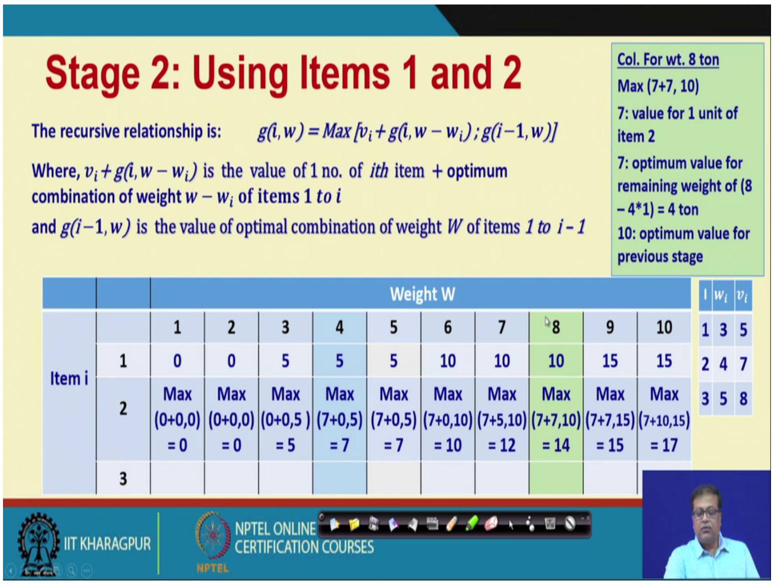
The 1 part is we are adding item 2 into the knapsack we are considering a knapsack of both items 1 and 2 while considering the previous value. So, the highlighted one should be what? It should be 5 can you say that because that is the optimal value of a 5 sized knapsack of a single item right of the item 1 is it all right. So, that is the previous optimal right and that 5 is here.

So, you see this is explained here that is the 5. So, 5 optimal value for the previous stage. Now, how this is computed; 7 what is 7, 7 is the value corresponding to the current item. So, current item i the optimal value is 7. So, this is that 7 right and what is this 0 the other value that is 0 is supposing these knapsack these particular item weight is 4 tons right.

So, total weight was 5 tons. So, 5 minus 4 is 1 ton. So, what is the you know knapsack size of 1 the optimal value was 0. So, that 0 we put here right. So, that is 7 plus 0 and 5 the you know optimal value therefore, should be 7 at the knapsack size of 5.

So, I hope you understood see if you have understood now you should explain how an 8 knapsack should be evaluated in the same stage that is items 1 and 2.

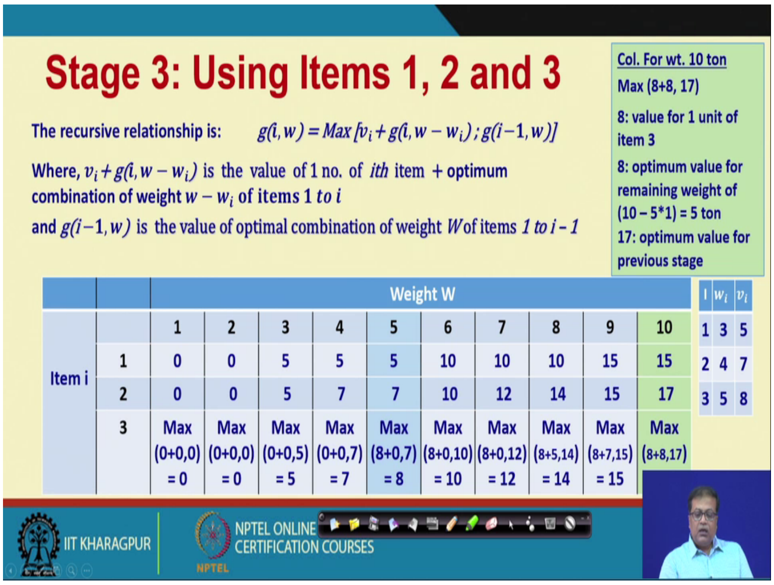
(Refer Slide Time: 26:55)



Why items 1 and 2? We are considering item 2, but we are using the optimal value at the item 1 level. Is it all right? So, once again what is the 7? 7 is the value for the item 2, these 10 is the previous value for the same knapsack size and since out of a 10 knapsack size we have taken out 4 tons because of a single item.

So, the 6 knapsack size sorry we are at the 8. So, 8 was the knapsack size.

(Refer Slide Time: 27:42)



So, 8 knapsack if you take out the 4 tons then that I am sorry once again. So, we have the 10, 10 is our knapsack size. So, out of these out of this 8 knapsack size if we take out the 4 tons that leaves us the remaining 4 tons and so, the other 4 tons will be you know the optimal of 7 and that is how we got this 14. Is it all right?

So, this is how is we calculate and one more example you see if I have a 10 knapsack size then we make use of the 6 knapsack because the weight is 4 tons. So, that optimal is 10, 7 is the current weight, 17 15 was previous. So, we get the 17 knapsack size.

So, that completes our item 1 and 2, let us go ahead with all the 3 items together. So, when we take all the 3 items together a similar calculation can be done. While we do a similar calculation we should keep in mind that at this stage we are considering all the 3 items that is 1, 2 and 3.

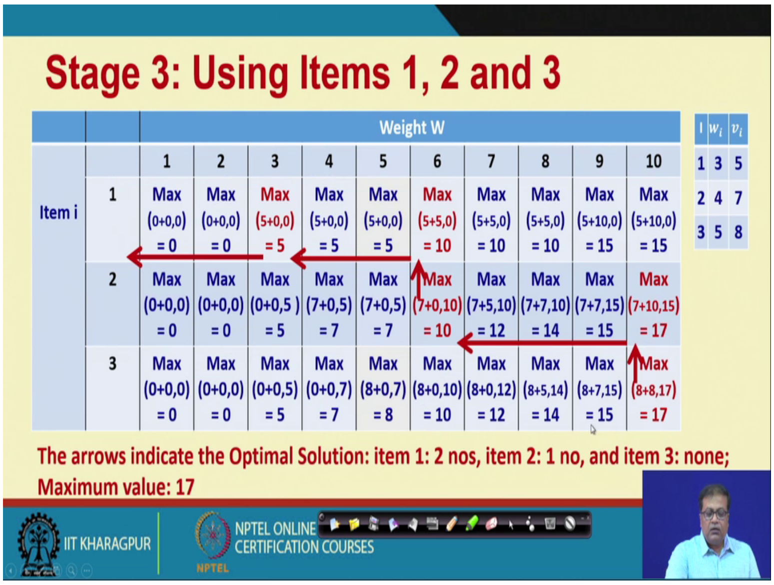
As we consider all the 3 items we make use of the stage 2 values which are considering both item 1 and item 2. So, if I add item 3 things now with the optimal value. So, previous stage optimal is g i minus 1 w is that of stage 2, I mean item 2 corresponding to stage 2 these optimal values are of stage 2 and those stage 2 values 10 really considers both item 1 and item 2.

So, when I add the item 3 value that is 8 to this you know it becomes the optimal value for the all the items together. So, if I have of knapsack size of 6 then the current weight is

5; obviously, the remaining knapsack will be one whose optimal value is 0. So, 8 plus 0 and the 10 that is previous we get 10 that is at the 6 knapsack size.

One more example, if we have a 10 knapsack size and we are here, 8 is the current value that 5 is the weight so, if I consider the 5 knapsack the optimal was 8, so, these 8 and 8 and 17 previous, so; obviously, the value should be 17.

(Refer Slide Time: 30:25)

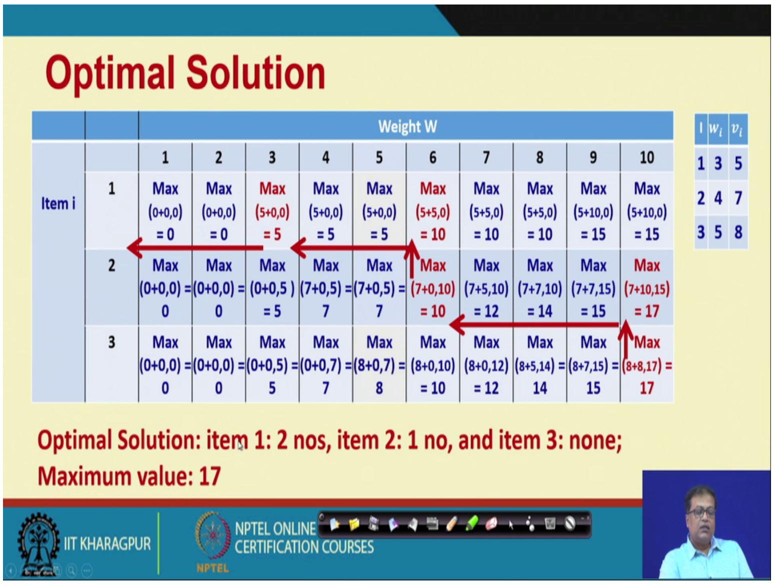


So, combining them all now, we have the total result and everything has been put here. We have still one job to do that is finding the optimal solutions. So, how do I find out optimal solution? We start at the highest possible value. Look here the highest possible value out of everything here that is the 17.

Now, the question is that where from this 17 came? See the 17 that came that 8 plus 8 was lower than 17; that means, the previous value was importance. So, we go to the previous value right. These 17 has come from you know there is the 10 this 10 because these 17, 1 item was taken of value 7 and we have gone to this stage.

These 10 again the previous value was higher so, we have come from here and these 10 has been obtained because the you know weight is 3 came from here that is 5 and this 5 has come from the 0th value. Obviously, so, what does it say? It say that have one item from these values one item of item 2, two items of item 1. So, what is our optimal solution? Item 1 2 numbers item 2 1 number and item 3 none and what will be the maximum value, 17.

(Refer Slide Time: 32:01)



So, let us combine them all once again that is the 17 value that we have got clearly shows that third item 17 is the highest knapsack value that is the optimal solution, but the value has not come from consideration of third item. It has come from previous item that is items 1 and 2. Now, again the 7 plus 10 is higher than 15; that means, at least 1 item 3 value has been taken. So, taking considering that 1 item if I leave out then the remaining knapsack that remains is that of 6 because 4 is the weight of the second item.

So, when we need do that then I see that the knapsack now of size 6, you know we have 7 plus 0 that is 10 and using this knapsack again it has come from previous value which has come from the consideration of 2 items. Is it all right? So, that is how we find the optimal solution and the maximum value using this algorithm right so.

Thank you very much.