Constraint Satisfaction Problems

Prof. Deepak Khemani
Department of Computer Science
Indian Institute of Technology – Madras

Module - 1, Lecture - 03

Okay so let's continue with examples of CSP. So one nice example which comes where CSP has been used quite effectively comes from the area of scene labelling. So what happens is that when you are processing some images and if you have line drawings, you may sometimes want to label these line segments using one of the four different possible kind of labels.

So the four different labels are as follows. So when I say four labels, essentially I'm talking about a particular class of objects which is called trihedral objects. Trihedral objects are objects where every vertex is made up of three edges coming together. Exactly three edges which means we can only talk about a particular kind of object essentially.

So an example of trihedral object is something like this. So the object is such that every vertex that you see in the line drawing is made up of three edges coming together which also means that there are three faces coming together, or three planar faces.

So we are only confining ourselves to very simple kind of objects here. But the real implementations have gone much beyond trihedral objects and you can have different kinds of, more complicated kinds of objects and even objects which have, for example shadows or cracks and things like that. We assume that our object is lit in such a way that the only lines that we see are the edges of these trihedral objects. So this is just an example of a trihedral object.

Now the four labels that we are talking about are, the four labels are of this kind that you can either label an object with an arrow pointing to the right or you can label it with an arrow pointing to the left, because given a line you can have the arrow on either side. Or you can label it with a +. Or you can label it with a -. So there are four different kinds of labels. And there are four different kinds of vertices that we can see in these objects.

What are the meanings of these labels? An arrow pointing to the right means you can see material on one side and material is always on the right, when you are looking in the direction of the arrow essentially, which applies to both these kinds of arrows. Basically on the top arrow, your material is below that line and in the bottom, second arrow, material is above the line. A + sign denotes a convex edge and a – sign denotes a concave edge. So you can see that each line can be labelled in four different ways.

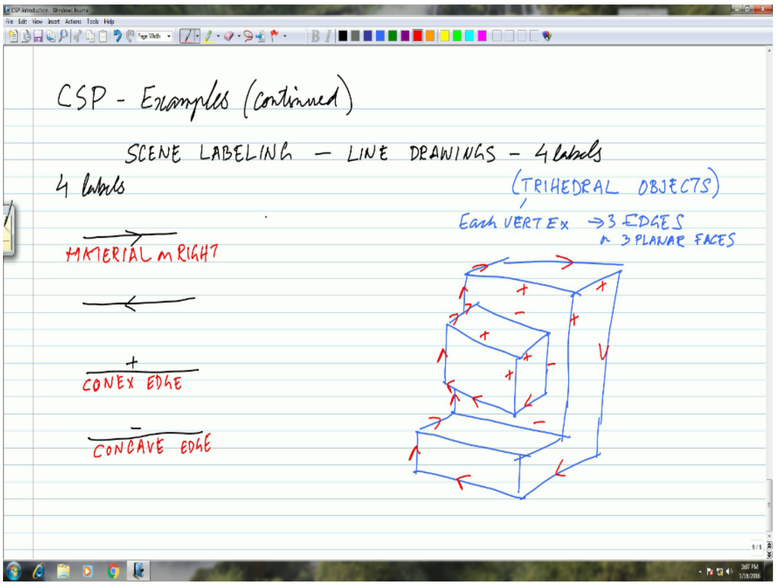
So let me illustrate these labels on the given line drawing that we have. So given that this is a solid object that we are talking about, first we can kind of label the outer most regions because we know that the material is on the inside. So we traverse in such a way that the arrows are oriented in such a direction that the material is now confined inside the arrows. And now inside this, for the rest of the figures we can see that this labels, so there are these three examples of these three convex edges that I'm labelling. And this is what we mean by a convex edge essentially, that if you were to look at the object, then the material would be kind of in a small area.

As opposed to that, we have negative edges. For example, this edge and this edge and this edge. These three edges that I have just coloured, they are negative edges because when you look at the angle made by the solid part of the object, its more than one eighty degrees and for convex edges the angle made by the solid part is less than one eighty degrees.

So this particular object would be labelled in the manner that I have just shown you. So this is +, this is +, this is +. Okay so we have labelled the entire object.

The task in scene labelling is to label a given object using one of these four labels. Now you can see that given the number of lines and with each line being possibly labelled in four possible ways, the number of combinations of labels is huge essentially. So what do we do is that we try to exploit the properties of these different kinds of labels. So for example if you look at, we identify vertices. So let me take a new page and see.

Refer Slide Time 6:53)



So we identify four types of vertices. Equal to three edges. And only some combinations of labels are allowed. And remember we are talking about trihedral objects here essentially.

So what are the different kinds of combinations? So one edge which we call as a Y joint or its also called a fork, has labels. So if I call this edge as a and this is b and this is c. Then the labels that I'm allowed to use are, that all three can be +. Or all three can be -. And you can try to imagine objects where these are consistent labels. Or one of the labels can have an arrow pointing towards it essentially.

So let me illustrate that here. So this can have an arrow, this can have an arrow and this can be negative. So you can imagine an object or maybe in a short while I'll draw an object in which we can have. So we have arrow, arrow, -. And this - can be any one of the three. So it can be here. Or it can be in the middle and so on. So this is a Y joint.

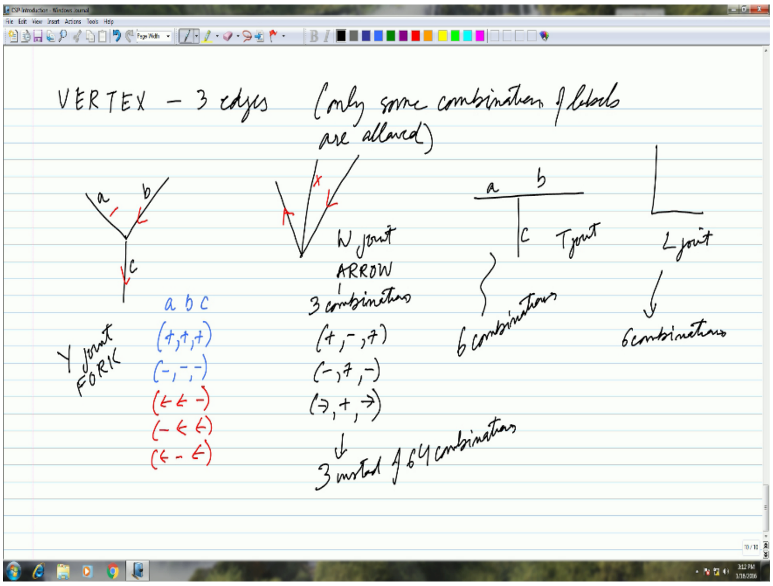
Then we have what is sometimes called as a W joint or an arrow and this has three combinations. So the three combinations are +, -, + or -, +, -. Or one of them can be an arrow. So for example if I was looking at a cube, then this would be an arrow, this would be an arrow and the one in the middle would be a + essentially.

Then the third kind of object is a T joint. So we label the three segments like this and you can imagine that this is part is like when you are looking at the table from the side or something like that. And this has six combinations.

And the last is an L joint. And the L joint also has six combinations. So we will not draw them here. But you can see that essentially the number of combinations, so instead of, when you have three lines coming together, you have in this W joint for example three instead of sixty-four combinations.

So only these three particular kinds of labellings are consistent if you are talking about solid trihedral objects. And you don't have to really try the sixty-four different combinations. Likewise, in these T joint and the L joint, only six combinations are allowed. So six out of sixty-four or six out of sixteen. And in the case of Y joint only five out of sixty-four are allowed. And even from this five, three of them are basically rotations of one another.

(Refer Slide Time 11:29)



Now, they all are very famous algorithm called waltz algorithm, which was used for labelling line drawings. And this kind of labelling is called Huffman Clowes labelling. These kind of labels have been based on the Huffman Clowes. Huffman. And what the waltz algorithm does is that it illustrates the idea of constraint propagation. We saw this idea when we were talking about cryptarithmetic puzzles. But waltz demonstrated this very effectively for line drawings.

So let me illustrate essentially. So supposing you are looking at a line drawing and you are looking at a fork like this and an arrow like this essentially. Now obviously this is part of a larger object essentially. Now the basic idea of constraint propagation is that whatever label I use here for this particular segment must be the same as the label that I use here because they are part of the same line object. We are talking about one particular object. And whatever is the kind we will use that.

And the second thing is that once we have labelled one object, so for example, in the W segment, if we have somehow managed to label this as a +, then we know that one of, the only option that is available to us now is that this must be a + and this must be a +. So these two are constraints because the set of constraints that we have said is that that's the only way. If you have labelled one of the edges in a W object with a +, then the others must be labelled like this.

So let me try to complete a figure with that. So this is a figure that we were looking at and this is a W object that we are looking at. So you can see that if the rest of the figure looks like this, then the labelling that I was talking about here is consistent. So this is + and this should be -. You can't have +, +, + for a fork. It has to be - here. And because this is +, it must be + at the other end. And then these two must be + and the rest can be labelled in a fashion.

So we saw we can say two instances of W joint. One is here with a -, arrow, arrow. And one is here with a +, arrow, arrow. And we can quite easily visualise this solid object.

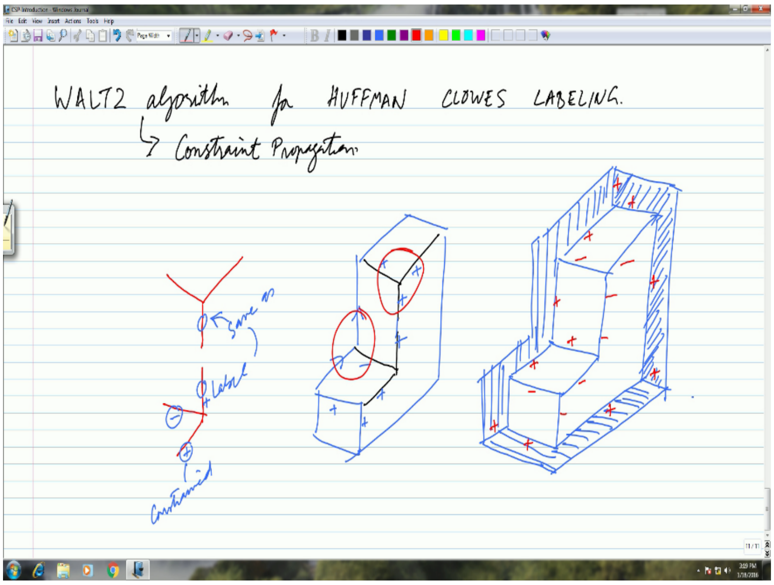
But just at the same time, just try to imagine if this was like this. Okay so let me see if I can draw this. The same kind of line drawing but with a different labelling. And that is because in this example it is part of a larger figure which looks like this. So if you can visualise this object. Okay this is the object that I had in mind, sorry. So I have added one extra layer of lines outside.

Now if I were to try and shape this for you to be able to visualise this object, so let's say this is one edge and this is another edge below and this is a third edge; third face I think. I don't know if you can visualise this but the labelling that I have in mind here is that this, for this same joint that we were talking about, the labels are different. Here it is -, - and -. So once I label this which means these are concave edges, you can now try to visualise this as a hollow object as opposed to a solid object here. And that's why the three labels are negative essentially.

Now obviously you can see that there are many different possibilities and the way that we go around solving this is by propagating constraints. Once we know the label of one vertex, then we can propagate it along common edges to other vertices and because each vertex has a small set, an allowed set, we can label that effectively essentially.

So this should be +, this should be -, this should be -, this should be -, this should be -, and this should be +, and so and this should be +, and this should be +, and this should be + and the rest is the boundary kind of thing.

(Refer Slide Time 17:09)



Let me take another example now. And this is an example of a crossword puzzle. So we are not only talking about solving puzzles but we are talking about setting crossword puzzles. So what is the task of setting a puzzle? That you are given some grid. So let us say we have this grid and typically in a crossword, some parts of this are shaded so you don't have to fill in letters there. And the task is to fill up the remaining part of the crossword with letters in such a way that every sequence of letters forms a meaningful word.

So you can imagine, you can see there is a five letter word here and there is a four letter word here and so on and this thing. But the thing is that the letters that you are, for example if you put in laser here, then this four letter word must be a meaningful word. So for example it could be same. Okay. And then you need a three letter word here. Maybe it is rat. Something like that typically.

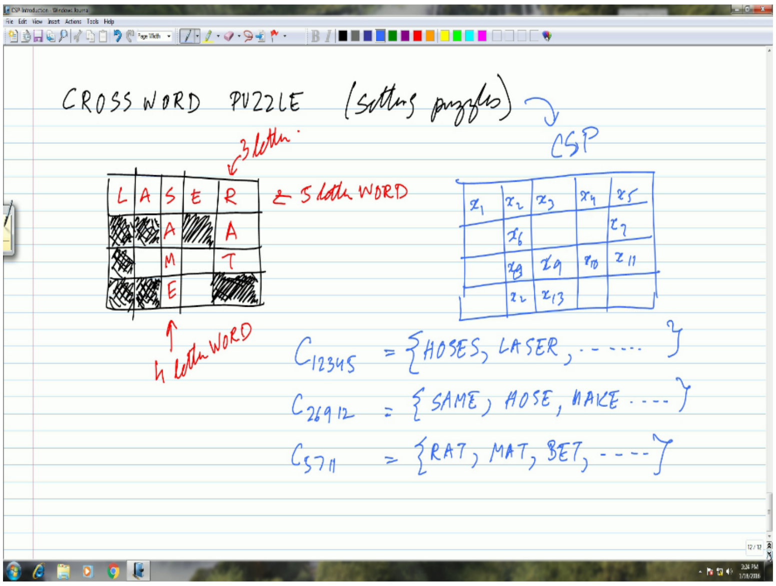
So how can we now pose this problem as a CSP? So one way to do it would be to say that every box where a letter has to be filled is a variable. So this is x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12, x13, something like that.

And now we can say that there is a constraint, so let's, let me name the constraint by the indices of this variable. So there is a constraint C-1-2-3-4-5 which means there is a constraint on the set of five letters that I can put and I have basically, whatever my lexicon is, I will choose five letter words from there. So for example hoses, or this laser and so on essentially. So basically the set of allowed words is the set of constraints.

So likewise I have a constraint on C-2-6-9-12 which is for the vertical word and I have a constraint on C-5-7-11 and so on. So these are all four letter words like same, hose, make, whatever the letters that are allowed to me are here. So this is three letter words like rat, and whatever, mat and bet.

So I can pose this as a constraint satisfaction problem. Basically it means that I can assign values to variables x1, x2, x3, x4, x5 only based on the constraint on the five variable essentially.

((Refer Slide Time 21:56)



Now I can alternately choose my words as variable essentially. Now what are the words? So I can fill in six words here essentially. Let me call this w1. So w1 is a five letter word which will fit here. Then w2 is a four letter word which will fit here. w3 is a three letter word which will fit here. Then w4is here. It's a four letter word. And w5 is here. It's a two letter word. And w6 is a two letter word.

So I can now say that I have six variables. And what are the constraints on those variables? So for example I can say that if w1 is equal to laser, then my w2 is going to start from here. So w2 must begin with s. So for example then w2 could be same or something like that. So these are the kind of constraints between these two kind of variables and we can draw them essentially.

Now the interesting this about this crossword puzzle is that if we draw the constraint graphs of the two formulations that we had then in the first formulation which you can see, there are thirteen variables and let me try to draw a constraint graph here.

So formulation one – x1, x2, x3, x4, x5, so remember that x1 is here. I'll just label them; two, three, four, five. And we had said that we draw a constraint graph between two variables if they participate in a constraint. Now these five variables x1 to x5 are participating in a constraint of five variables which means that we must now make a clique here. We must connect every edge with every other edge. And therefore we have a part of the constraint graph here essentially.

Now if you look at this variable x3, x3 is also interacting with x6, then this is seven, then this is eight, nine, ten, eleven, twelve, thirteen. So x3, which we had drawn here is also interacting with x6 and x9 and x12. So that will form another clique here and they will be connected like this.

So likewise x5, we've drawn x5 here. x5 interacts with x7 and x11 and they form another clique. Likewise, we can draw the constraint graph for the rest of the diagram essentially.

So in some sense we say that this one is a hyper edge. And it's also, these five variables are the scope of the constraint that we called as C-1-2-3-4-5. So if you want to draw the constraint graph in the original first formulation, then we would have these thirteen variables and we'll have these different connections between them essentially.

Whereas we have this notion of a dual. So for every constraint, for every CSP, we have a dual CSP. And we construct a dual CSP as follows that the variables are the scopes in the primal CSP. So this first one was the primal CSP. And edges are labelled with common elements in the scope.

So for example, if I take the scope of this C-1-2-3-4-5 and I can without loss of generality call this as w1. I'll call this variable w. I can call it C-1-2-3-4-5 or I can call it w1 but w1 is kind of consistent with the second formulation that we had done essentially.

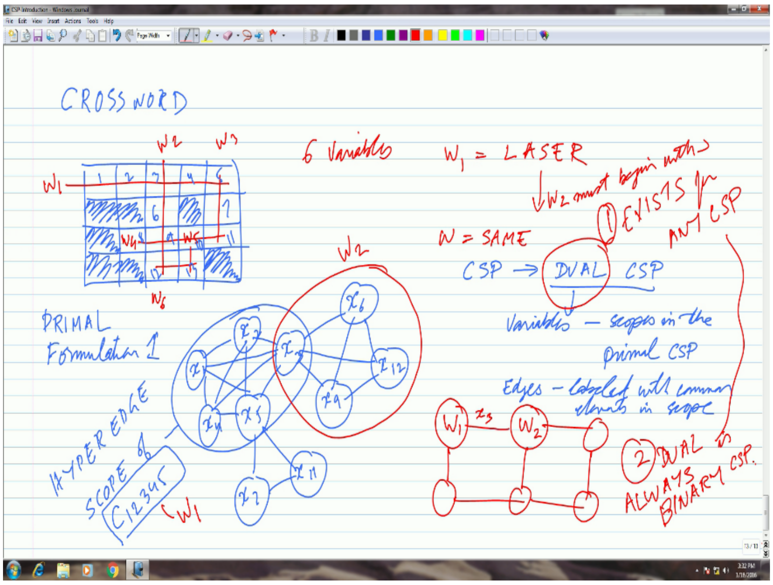
So I have w1 here. And then if I look at another such scope which is called w2, then I have w2 here. And what is the label? The label is x3. Whatever was shared between these two constraints, these two scopes is the label of this edge essentially.

So if you now fill up the rest of the details you will see that the constraint graph of the dual in this example happens to be identical to the constraint graph of the second formulation that we did which is with six variables.

The interesting thing is that the dual exists for any CSP, that's one thing and the second thing is the dual is always binary. So here is an interesting observation you can make that we can generalise form here which says that if you are given any CSP. Now if you see when we did the first formulation of this crossword puzzle, we had constraints which had arity four and five and three and two and so on but we could convert that CSP into its dual and the dual is constructed by taking the scope of every one of the constraints in the primal and making it a variable in the dual and the edges become what are common between the scopes of those variables. And it turns out that dual is always a binary CSP. So we have a nice mechanism for converting any CSP into a binary CSP and if we have some well-defined techniques for binary CSP then we can use them to solve those problems.

Okay so I will stop here with these examples and there are many other examples that you can look up in different sources, for example, in Rina Dechter's book you can find a couple of more examples. But we'll stop here with examples and we'll move on to the rest of the course which involves solving CSPs essentially. So in the next class we will basically assume that we have a CSP and we'll start with binary CSPs maybe and then try to find algorithms which can be used for solving them essentially.

(Refer Slide Time 30:03)



CSP Introduction - Windows Journal
File Edit View Insert Actions Tools Help
CROSSWORD
W1
W2
W3
W4
W5
W6
6 Variables
W1 = LASER
W2 must begin with S
W = SAME
1 EXISTS for ANY CSP
CSP → DUAL CSP
Variables — scopes in the primal CSP
Edges — labeled with common elements in scope
PRIMAL Formulation I
HYPEREDGE
SCOPE of C12345
2 DUAL is ALWAYS BINARY CSP.