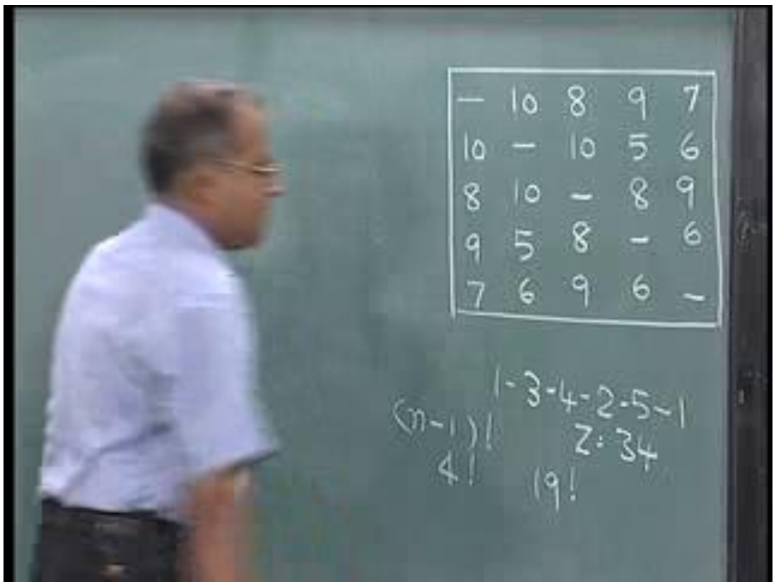
**Advanced Operations Research**
**Prof. G. Srinivasan**
**Department of Management Studies**
**Indian Institute of Technology, Madras**
**Lecture minus 26**

# Heuristics for TSP

In this lecture, we continue our discussion on the Travelling Salesman Problem. We look at heuristics to solve the Travelling Salesman Problem. First we have to look at what a heuristics is and then, we try and explain why we provide heuristics solutions to the Travelling Salesman Problem. The first question is what is a heuristics solution to this problem? In the earlier lectures we had seen 3 types of branch and bound algorithms to solve the TSP. All of them gave us the exact solution which is shown here.

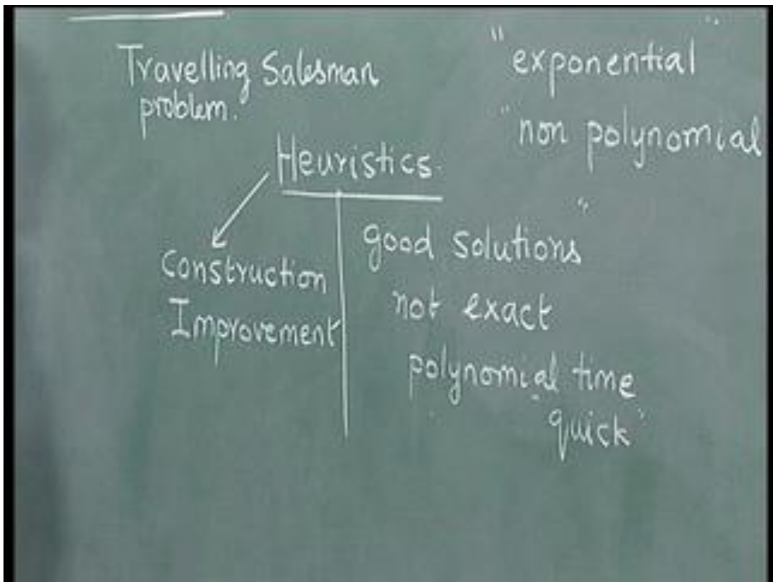
(Refer Slide Time: 01:00)



The exact solution being 1-3-4-2-5-1 with Z equal to 34; this exact solution tells us that the best value or the minimum cost solution is 34 and the person travels from 1 to 3, 3 to 4, 4 to 2, 2 to 5, 5 to 1 with Z equal to 34. All above branch and bound algorithms were able to give us this value with Z equal to 34. The 3 branch and bound algorithms we had solved, using this example of a 5 - city Travelling Salesman Problem. We have already seen that, for n city problem, we have n minus 1 factorial feasible solution, out of which the best one is with 34. In some for the 5 - city problem, we have a maximum of a 4 factorial feasible solutions and if we look at a twenty - city problem, then we might even go to as high as nineteen factorial feasible solutions.

When we apply the branch and bound algorithm from any one of the 3 methods that we have seen let us assume that we would choose the third branch and bound that we saw in the earlier lecture. When we apply this branch and bound algorithm to large problems, large meaning of the order of hundred cities or more, in fact hundred is not seen as a very large problem, but we apply it to a hundred city problem. Then we will observe that the number of nodes is very high, the CPU time taken is also very high. It may not be able to solve a hundred factorial problems within a reasonable computing time or within a reasonable CPU time. That happens because of the exponential nature of the Travelling Salesman Problem, now this leads us to a discussion on the complexity theory.

(Refer Slide Time: 03:18)

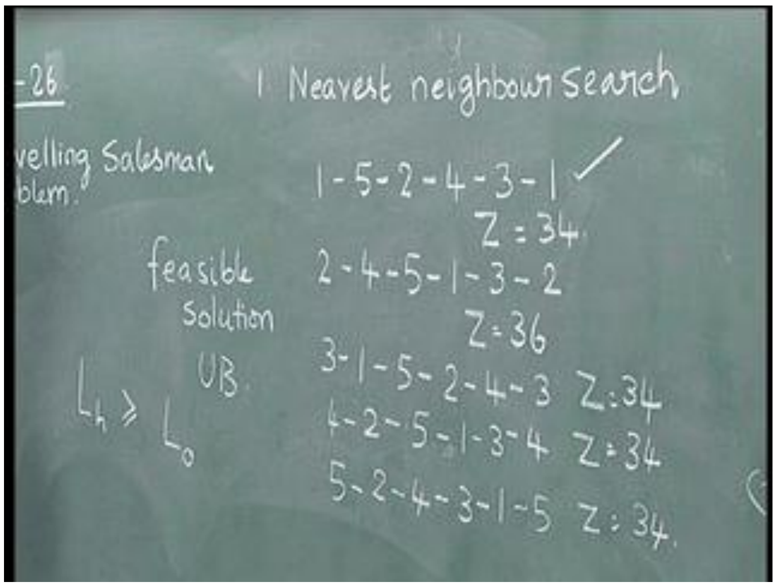

Travelling Salesman problem.
"exponential"
"non polynomial"
Heuristics
Construction
Improvement
"good solutions"
not exact
polynomial time
quick"

The branch and bound algorithm that we have seen comes in the category of an exponential or a non polynomial algorithm, where the computational effort taken is not described as a polynomial function of the problem size. Any branch and bound algorithm in the worst case, can evaluate all possible solutions, which means for n city Travelling Salesman Problem, it might end up evaluating n minus 1 factorial solution which is an exponential number. For whatever problems when we do not have algorithms, that have what is called a polynomial complexity branch and bound is a non polynomial complexity. Wherever we have algorithms that do not run in polynomial time then it becomes necessary to get good solutions not necessarily exact but to get good solutions with algorithms that run in polynomial time. Heuristics algorithms actually provide us with all these 3 things. Every heuristic algorithm attempts to find a good solution to the problem, does not guarantee exactness, sometimes we may get the exact solution, many times we may not. Most importantly, it gets quick solutions in terms of computational time and effort.

It is polynomial time and effort and they are mostly quick with respect to providing these solutions. While branch and bound might take and exponential or a non polynomial amount of time to terminate every heuristic algorithm terminates within polynomial time and provide quick solutions or good solutions and close to exact solutions though not exact solutions all the time. Therefore, it is necessary to look at heuristics for the Travelling Salesman Problem because so far, all the exact algorithms that had been developed for the TSP run in exponential time. It is necessary to look at heuristics algorithms that do not provide exact solution or do not guarantee exact solutions, but provide good solutions that run in polynomial time. There are different categories of heuristic algorithms or heuristic methods. Some of them are called construction heuristics and the others are called improvement heuristics.

In this lecture series we will see some construction heuristics as well as some improvement heuristics to the Travelling Salesman Problem. Let us start the discussion with a first heuristics.

(Refer Slide Time: 06:50)



-26

velling Salesman
blem.

1 Nearest neighbour search

1-5-2-4-3-1 ✓
Z = 34

feasible solution

2-4-5-1-3-2
Z = 36

UB.

3-1-5-2-4-3 Z = 34

$L_h \geq L_0$

4-2-5-1-3-4 Z = 34

5-2-4-3-1-5 Z = 34.

The first heuristics that we will describe is called the nearest neighbour search. Let us provide a quick solution to the Travelling Salesman Problem. Let us assume that the person starts at city one, when the person is at city one, we have to find out which is the next best place to go. thus, 1 to 2 is 10, 1 to 3 is 8 9 and 7. We would assume safely that the person would go to the nearest neighbour which means that place or city which is closest to where the person is at present. From 1, this person would go to 5 because 1 to 5 is a smallest number here. From 5, now the person can go to either 5 to 2 or 5 to 4. It could be any one of them. Let us assume the person goes from 5 to 2. From 2 it is 10, 10 5 and 6. The smallest number being 5, the person will go to 4 and from 4 the smallest number is 2. But the person will not come back to 2 because 2 to 4, 4 to 2 will be a sub - tour.

The person will try the next one which is 5 again. He does not come because 5 to 4 and back to 5 is a sub - tour. The only thing that is available is 3 and then the person comes back to 1. This is how the nearest neighbour works. You start with any city, preferably the first and then find out the nearest city. If the nearest city does not give a sub - tour, then add it. Otherwise find out the next nearest and repeat till you do not you identify a city which does not result in a sub - tour. Add it and repeat the procedure till all cities are covered. This will be the nearest neighbour solution, when we start with city one. The value for this, let us call it as Z, 1 to 5 is 7, 5 to 2 is 6, 13, 2 to 4 is 5, 18, 4 to 3 is 8, 26, 3 to 1 is 8, 34. Let us do the nearest neighbour with city 2. We start with city 2 and then from city 2, the person will go to city 4 because 2 to 4 is the smallest. From 4, the person will go to 5 because 4 to 2, the person will not do because it is a sub - tour. From 4 he will go to 5 and then from 5 the person will go to 1. Then the person will go to 3 and then come back to 2. This will give Z equal to 2 to 4 is 5, 4 to 5 is 5, plus 6 equal to 11, 5 to 1 is plus 7, 18, 1 to 3 is 18 plus 8 26, 26 plus 10 is 36. Repeat, 2 to 4 is 5, 4 to 5 is 5 plus 6 11, plus 7 18, 18 plus 8 equal to 26, 26 plus 10 is 36. So, different starting cities can give different solutions.

We have already shown solutions when we start with city one and city 2. We may do it with city 3, we may do it with city 4, and we may do it with city 5 and then choose the best out of these 5 possible solutions as the best one. If we may do it with 3, we would start with 3, from 3 we would come back to 1 and from 2 we will go to 5. From 5 we can go to either 2 or 4. Go to 2 and then 4 and then 3, with Z equal to 3 to 1 is 8, 1 to 5 is 8 plus 7 is 15, plus 6 equal to 21, 21 plus 5 equal to 26, 26 plus 8 is 34. From 4, we would do 4 to 2 is a smallest 2 to 5, 5 to 1, 1 to 3 and 3 to 4 with Z equal 5 plus 6 11, 5 to 1 7, 18, 18 plus 8 26, 26 plus 8 equal to 34.From 5, we will do 5 to 2, 2 to 4, 4 to 3, 3 to 1 and 1 to 5 with Z is equal to 34. 5 to 2 is 6, 2 to 4 equal to 6 plus 5 equal to 11, 4 to 3 11 plus 8 equal to 19, 3 to 1 19 plus 8 equal to 27 plus 7 equal to 34.

Let me a repeat, 5 to 2 is 6, 2 to 4 is 6 plus 5 is 11, 4 to 3 11 plus 8 equal to 19, 3 to 1 19 plus 8 equal to 27 and plus 7 34. Out of these 5 solutions, we realize that 4 of them give value 34 and 1 gives value 36. We also realise that 1 5 2 4 3 1 is different from 2 4 5 1 3 2, but if we look at this solution, 1 5 2 4 3 1, is same as 3 1 5 2 4 3 and so on. Effectively all these 4 happen to be the same solution. This is different. The best of them we could choose any one of them with value Z equal to 34, is the solution obtained using the nearest neighbourhood search. We say that this could be the solution obtained using the nearest neighbourhood search. The nearest neighbourhood search starts like this. We can start with any one of the cities, say city one. Find out the nearest neighbour, if the nearest neighbour is already there in the solution, leave it and now find out that neighbour which is not yet in the solution. This means it will not form a sub - tour.

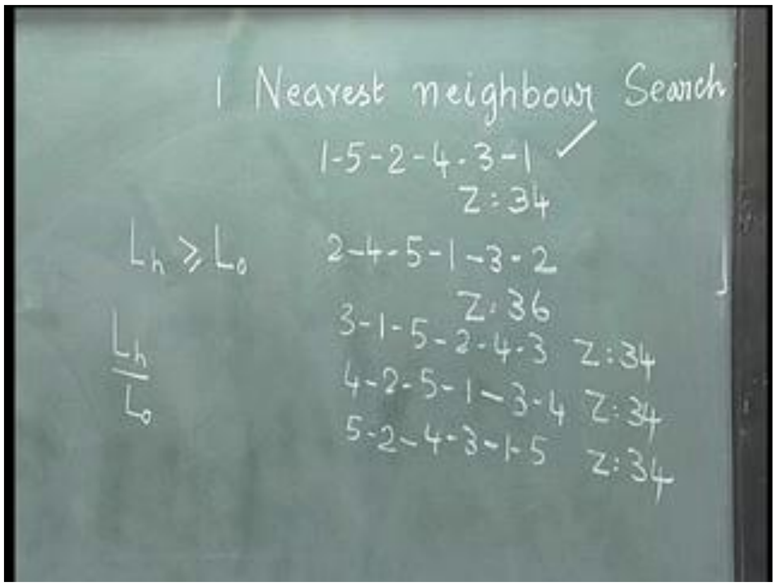
Add the neighbour and proceed till all the cities are exhausted. We come back to the starting point; repeat this for every city as we have shown here. So with one, with 2, with 3, with 4, with 5 and choose the best. This is an example of a construction heuristics. The reason is, we construct, we start from one and we slowly build or construct the heuristic, till we get a feasible solution to the problem. Every one of these is a feasible solution to the problem, just because we know already we have solved this problem, before we know that 34 is the best answer. We may say that this nearest neighbourhood search has actually given us the optimum.

We are able to make that statement because we know the exact solution or the optimal solution. Ordinarily, we would not be able to say that we have reached the optimum. The best out of these possible ones is the heuristic solution that is given by this particular heuristics. Each of these heuristic solutions is a feasible solution. Each of these is a feasible solution and actually provides an upper bound to the optimum.

The one thing we may be able to say is, the best of these being 34, 34 is an upper bound to the optimum. It is customary to use a notation if $L_0$ is the length of the optimal tour to the Travelling Salesman Problem. Then $L_h$, every solution that is obtained by this $L_h$ is greater than or equal to $L_0$, that is the only thing we might say. In this particular instance, we may be able to say that we have got the optimal. That is because we know that 34 is the optimum. We can quickly understand that the computational effort that is required to do this is far less than the computational effort that was required to do this branch and bound. More importantly, it is possible to show that we only do a polynomial number of steps to get each one of these solutions. This would qualify as a heuristic because it runs in polynomial time, it provides a feasible solution and it provides a quick good solution to the problem.

The goodness of any heuristics has to be measured. At the moment, we may be able to say that the best value is 34 but how good is that 34? We know that this $L_h$ being 34 let us assume that we do not know this $L_0$. What we now know is that $L_0$ is less than or equal to 34.

(Refer Slide Time: 16:43)



1 Nearest neighbour Search

1-5-2-4-3-1 ✓

Z : 34

$L_h \geq L_0$ 2-4-5-1-3-2

Z : 36

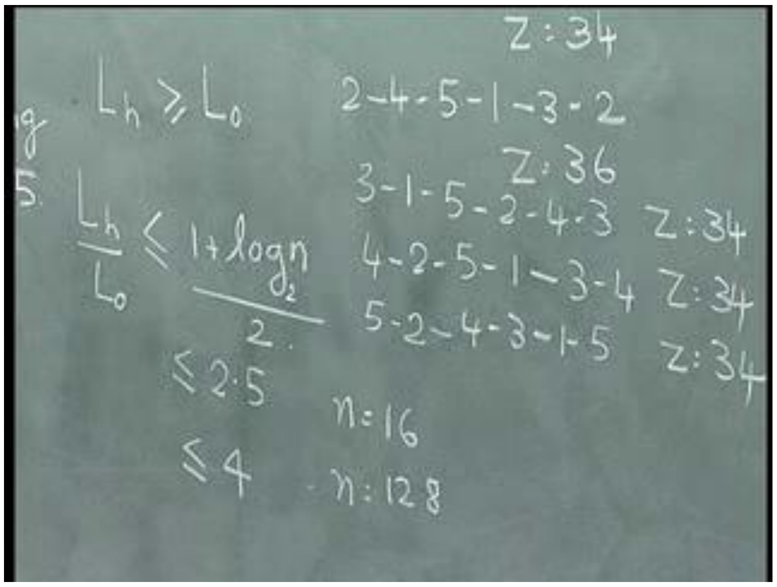
3-1-5-2-4-3 Z : 34

$\frac{L_h}{L_0}$ 4-2-5-1-3-4 Z : 34

5-2-4-3-1-5 Z : 34

$L_h$ represents the length of the heuristics solution; $L_0$ represents the length of the optimal solution. So $L_h$ is always greater than or equal to $L_0$ because it is a heuristic. We need to evaluate the goodness of this heuristic. To find out how good this heuristic solution is, with respect to the optimum. We can derive some expressions for $L_h$ by $L_0$ which tells us the extent to which the heuristic solution can be bigger than the optimal solution. Such an analysis can be carried out. The expression for $L_h$ by $L_0$ for the nearest neighbourhood search is given by this. $L_h$ by $L_0$ is less than or equal to 1 plus log n to the base 2 by 2, where n is the problem size and the logarithm is 2, the base 2.

(Refer Slide Time: 17:53)



Z : 34

$L_h \geq L_0$ 2-4-5-1-3-2

Z : 36

3-1-5-2-4-3 Z : 34

$\frac{L_h}{L_0} \leq \frac{1 + \log_2 n}{2}$ 4-2-5-1-3-4 Z : 34

5-2-4-3-1-5 Z : 34

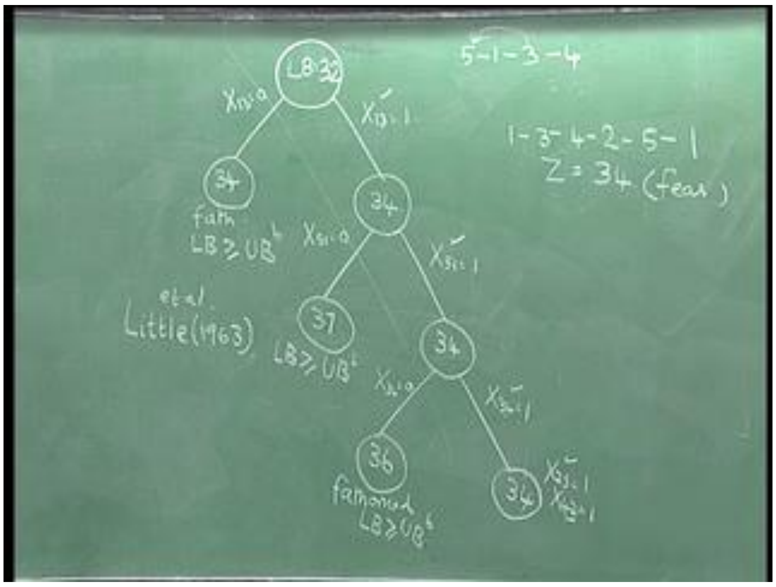
$\leq 2.5$ $n = 16$

$\leq 4$ $n = 128$

This shows that $L_h$ by $L_0$ is less than or equal to a quantity greater than 1. We are not going to derive the expression for $L_h$ by $L_o$ here. However for later heuristics, we will derive expression for $L_h$ by $L_o$. What does this expression tell us? For example, if n is equal to say 16, then 2 to the power 4 is 16. So this will become less than or equal to 2.5, so 4 plus 1 5 by 2. For example, if n equal to 128, 2 to the power 7 is 128. So this will become less than or equal to 4. This shows that if we have a matrix with size 128, length of the heuristics solution will be within four times the length of the optimal solution. We also observe from this expression that, as n increases, the goodness of the heuristic comes down. We also need to look at heuristics, where either the goodness of the heuristic remains the same, irrespective of the length. Or we are able to get better expressions for this $L_h$ by $L_o$.

Those heuristics that we subsequently are going to see, heuristics such as twice around the tree heuristic and a heuristic using perfect matching, we will derive expressions to show that $L_h$ by $L_o$ is less than or equal to 2. In this case, irrespective of the size of the matrix and $L_h$ by $L_o$ not is less than 1.5 in this case, irrespective of the size of the matrix. In summary, the nearest neighbourhood search is a very simple algorithm. It stars with a vertex, it goes through a set of all other vertex that are nearest, which are not yet covered. Then it picks up the best solution from among these. The nearest neighbourhood search has a complexity of 1 plus log n to the base 2 divided by 2. In fact, the second construction algorithm that we will see is actually from here.

(Refer Slide Time: 20:35)

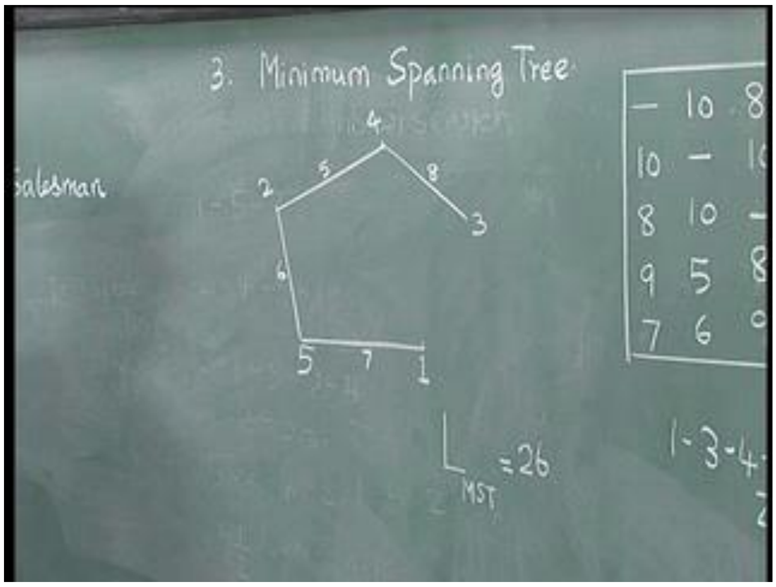


This is a branch and bound tree that we saw in the previous lecture, which was used to get the optimal solution. Somewhere during the previous lecture we even mentioned that this branch and bound can actually help us get even a heuristics solution. Let us go back to this branch and bound and let us say we started with 32. Then we branched here to get 34, then we branched here again and we got 34. We branched here and here we found out a feasible solution with Z equal to 34. This was a feasible solution with Z equal to 34. The moment we got a feasible solution here with Z equal to 34 for this particular example, we were able to fathom this, this and this. There were no more nodes that were available and we said that this is the optimal solution.

If for a particular example, that one of these values happens to be, say, less than 34 and then I said that we have to proceed from here till we get to the exact solution. If we want use this as a heuristic then right here we have a feasible solution with Z equal to 34. We could have just stopped here and said this solution by itself is a feasible solution. Therefore, this part of the branch and bound, up to the point where we get the first feasible solution, can be used as a heuristic solution.

The second construction algorithm or construction heuristic that we will see, is actually this portion of the branch and bound tree. When we come to this point, we get a feasible solution and this feasible solution acts as a heuristic to the Travelling Salesman Problem. Again, we would get show this as a heuristics solution. We may even ignore all the left side nodes, if we were using this only as a heuristic solution. The third heuristic solution that we will see is this.

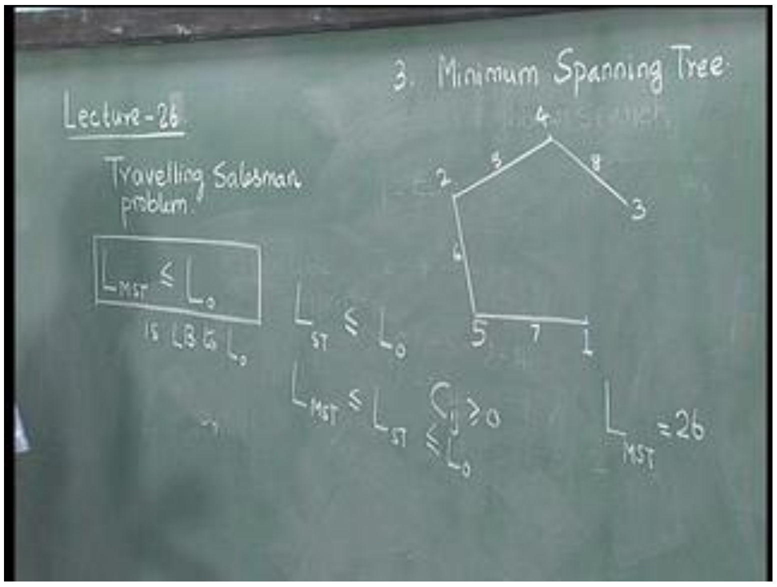
(Refer Slide Time: 22:50)



Let us apply the algorithm to try and get the minimum spanning tree associated with this matrix. We have already seen in earlier lectures that the Prim's algorithm or the Kruskal's algorithm can be used to get the minimum spanning tree. Let us first get the minimum spanning tree. The smallest one is between 2 and 4. We could start with 2 and 4 with value equal to 5, which is the smallest. In order to do with Prim's algorithm, now we go back we have put 2 and 4 here. We have 1 3 and 5 here.

2 to 1 is, 10, 2 to 3 is 10; 2 to 5 is 6, so minimum is 6. 4 to 1 is, 9, 4 to 3 is 8, 4 to 5 is 6. We could take either 2 to 5 or 4 to 5 and then do it. We take 2 to 5 with value 6 here as the next one. Now we go back, update this to 5 and bring this down to 1 and 3. We have to find out which one is going to join the minimum spanning tree. 2 to 1, 2 to 3 10 and 10 minimum is 10. 4 to 1 4 to 3 minimum is 8. 5 to 1 5 to 3 minimum is 7. So 5 to 1 come here with value 7. 1 gets updated here, 1 gets updated here and 3 come here. Now 2 to 3 is 10, 4 to 3 is 8. So minimum is 8. 5 to 3 is 9; minimum continues to be at 8 and 1 to 3 is 8. We could either do 4 to 3 or 1 to 3. Let us say we do 4 to 3 with 8. This is a minimum spanning tree associated with this. This has a length equal to 8 plus 5 equal to13 plus 6 equal to 19 plus 7 equal to 26. I am going to called this as LMST equal to 26. The length of the minimum spanning tree

associated with this matrix is 26. What does this information give us? We have solved a minimum spanning tree and how is this related to the Travelling Salesman Problem?
(Refer Slide Time: 26:02)



Let us take any feasible solution to the Travelling Salesman Problem. Let us take a solution 1 to 2, 2 to 3, 3 to 4, 4 to 5, 5 to 1. If we take this feasible solution now, 1 to 2, 2 to 3, 3 to 4, 4 to 5 and 5 to 1. This is a feasible solution to the TSP with L equal to 1 to 2 is 10, 2 to 3 is 10, 20 plus 8 is 28, 28 plus 6 is 34, 34 plus 7 is 41. The first thing that we observe is this L is greater than or equal to $L_0$ because this is a feasible solution to the Travelling Salesman Problem. $L_0$ is the optimal solution to the Travelling Salesman Problem. Every feasible solution to the TSP has 5 edges or n edges while the minimum spanning tree always has n minus 1 edge. If we remove any edge from this, any one edge from this, we may remove this, then we have 3 to 4, 4 to 5, 5 to 1, 5 to 2, we have a spanning tree.

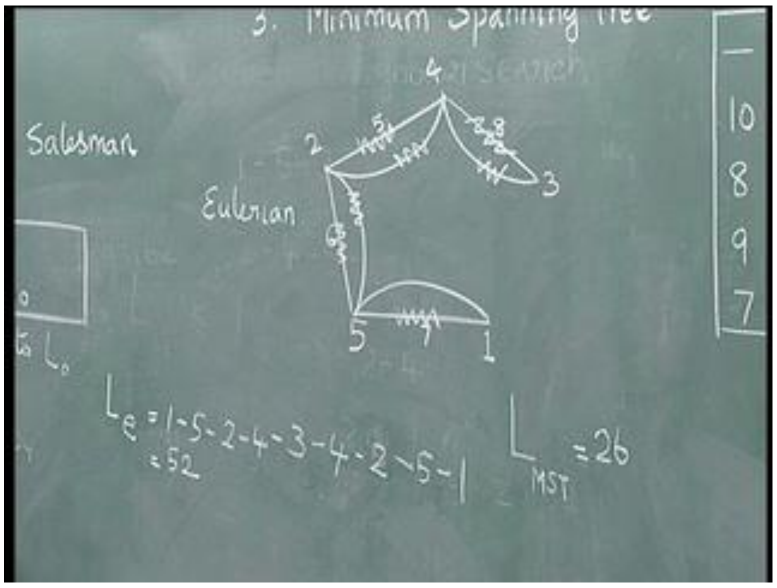
This is a spanning tree because this has 5 vertices, 4 edges connected. There is a path form every vertex to every other vertex. The original matrix has 5 nodes, this also has 5 nodes. So this is a spanning tree to this. Removal of every edge for any one edge from any heuristic solution gives us a spanning tree.

Let us assume that we do not know the optimal solution but the optimal solution also is made up of 5 edges. Therefore, removing one edge from the optimal solution would give us a spanning tree. If there is an optimal solution, which if the optimal solution has a length $L_0$ and if we remove an edge from this $L_0$, we get a spanning tree, so let us call that L as L spanning tree or $L_{ST}$. The length $L_{ST}$ should be less than or equal to $L_0$, not because $L_{ST}$ is 1 edge less than $L_0$. As long as all these $C_{ij}$s are greater than or equal to 0, which means we do not have any negative numbers coming here. This is a very reasonable assumption because this is either a cost matrix or a distance matrix or a time matrix, which means to go from i to j, we will only have to travel non - negative distances, non - negative times or non - negative costs.

As long as these $C_{ij}$s from this are greater than or equal to 0, every edge in $L_0$ is greater than or equal to 0. Therefore, removal of 1 edge from $L_0$ would give $L_{ST}$ which is less than or equal to the length $L_0$. This is a spanning tree to this matrix and this is a minimum spanning tree to this matrix. By definition LMST, length of minimum spanning tree, has to be less than or equal to length of any spanning tree, because by definition, this is the spanning tree with minimal length. We have $L_{ST}$ less than or equal are $L_0$ L minimum spanning tree ($L_{MST}$) less than or equal to $L_{ST}$, which is less than or equal to $L_0$. Therefore, we could say that the $L_{MST}$ is less than or equal to $L_0$ and length of the minimum spanning tree is therefore, a lower bound to $L_0$.

Here we may say that this $L_{MST}$ of 26 is a lower bound to $L_0$ and the optimum solution cannot be 26 or less. This is the relationship between the minimum spanning tree and the Travelling Salesman Problem. $L_{MST}$ is less than or equal the $L_0$ and it is a lower bound to $L_0$, is a very important result that relates the minimum spanning tree to the Travelling Salesman Problem. Having obtained the minimum spanning tree, what we can do now is, we duplicate the edges of the minimum spanning tree.

(Refer Slide Time: 30:55)



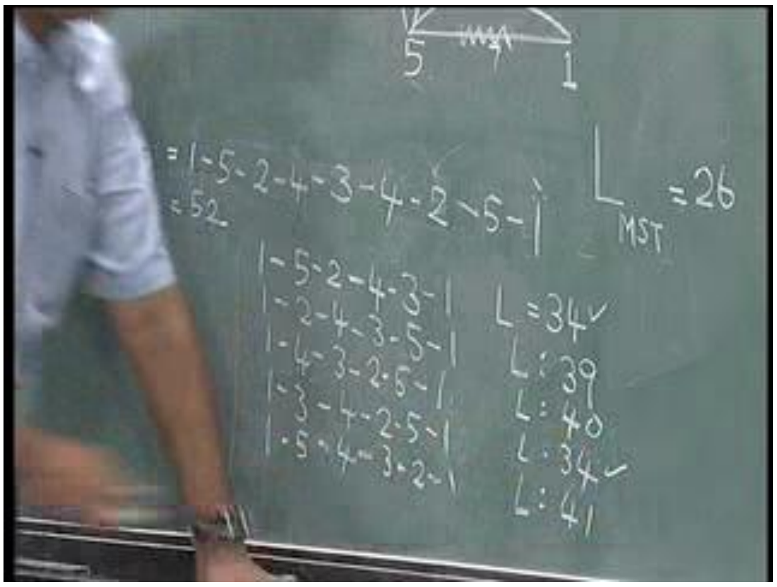
Then, try to write down the duplicated edged minimum spanning tree this way. For example I start from 1- 1 5 2 4 3 1 to 5. I am just marking that I have consumed this edge, so 1 to 5, 5 to 2, 2 to 4 and 4 to 3. Now I come back, I do 3 to 4; I come back and do 4 to 5.

I come back and do 4, 1 to 5, 5 to 2, 2 to 4, 4 to 3, 3 to 4, 4 to 2, 2 to 5 and 5 to 1. This way, I have covered all the eight edges of this. Remember that, the minimum spanning tree will have 4 edges and when we duplicate it, we would get eight edges. This is the length of twice the minimum spanning tree and I am going to introduce a new term. I am going to say that this particular graph, where we have duplicated the edges of the spanning tree, is an Eulerian graph. Therefore, I am going to call this with a e and I say $L_e$, length of the Eulerian will be the length of this, which will be twice the spanning tree which is 52, because this is double the length of the minimum spanning tree.

What is this Eulerian graph or Eulerian circuit? We will quickly see the definition of this an Eulerian graph is one where you can start with any vertex, go through every edge or arc once and only once and come back to the starting vertex. Which is what is did we went through 1 to 5, 5 to 2, 2 to 4, 4 to 3, 3 to 4, 4 to 2, 2 to 5, 5 to 1. It is also very easy to follow that, if we take a connected graph and we duplicate the edges, we can always get an Eulerian circuit. We can also understand that if we have a graph where the degree of every vertex is even, then it becomes an Eulerian graph.

What is a degree of every vertex? The degree of every vertex is the number of arcs or edges that are incident on this. If we take this, there are 2 which is an even number here. There are 4 here, there are 4 here, there are 4 here and there are 2. This happened because, we duplicated the edges. So if you take a given graph and duplicate all the edges, we will get an Eulerian. This Eulerian will have a length of twice that of the minimum spanning tree. This is going to help us in some ways to get feasible solutions to the Travelling Salesman Problem. How are you going to do that?

(Refer Slide Time: 34:18)



Let us start with this 1 and let us move. 1 to 5, 5 to 2, 2 to 4, 4 to 3 and if I come back from 3 to 4, I get a sub - tour; again I get a sub - tour so I do 3 to 1. I do 1 to 5, 1 to 5, 5 to 2, 2 to 4, 4 to 3 and back 3 to 1. This is a feasible solution to the Travelling Salesman Problem. This has L equal to 1 to 5 is 7, 5 to 2 equal to 7, plus 6 equal to 13, 2 to 4 13 plus 5 equal to 18, 4 to 3 18 plus 8 equal to 26, 3 to 1 26 plus 8 equal to 34. Is this the only feasible solution we can get? No, we can get many more feasible solutions. We can do another thing here; you could start with 1 to 5, 1 to 2, 2 to 4, 4 to 3, 3 to 5 and 5 to 1. You could do 1 to 2, 2 to 4, 4 to 3, 3 to 5 and 5 to 1. This is another solution you can get. Remember what we do is, we keep moving like a nearest neighbour but we look at several different solutions and make sure that we create tours as we move along. So 1 to 2 is 10, 2 to 4 is 5, 15, 4 to 3 is 8, 23, 3 to 5 23 plus 9 equal to 32 and 5 to 1 is 32 plus 7 is 39.

Repeat, 1 to 2 is 10, 2 to 4 is 5 15, 4 to 3 is 8 23, 3 to 5 9 32, 5 to 1 39. We can do one more solution like this. We could do 1 to 4, 4 to 3, 3 to 2, 2 to 5 and 5 to 1. Now 1 to 4 is 9, 4 to 3 is 9 plus 8 equal to 17, 3 to 2 17 plus 10 equal to 27, 2 to 5 27 plus 6 equal to 33, 5 to 1 33 plus 7 equal to 40. So we got another solution with 40. We can do one more here. We could do 1 to 3, 3 to 4, 4 to 2, 2 to 5, 5 to 1, 1 to 3, 3 to 4, 4 to 2, 2 to 5, 5 to 1. 1 to 3 is 8, 3 to 4 is 8 plus 8 16, 4 to 2 16 plus 5 21, 2 to 5 21 plus 6 equal to 27, 27plus 7 so L is equal to 34.

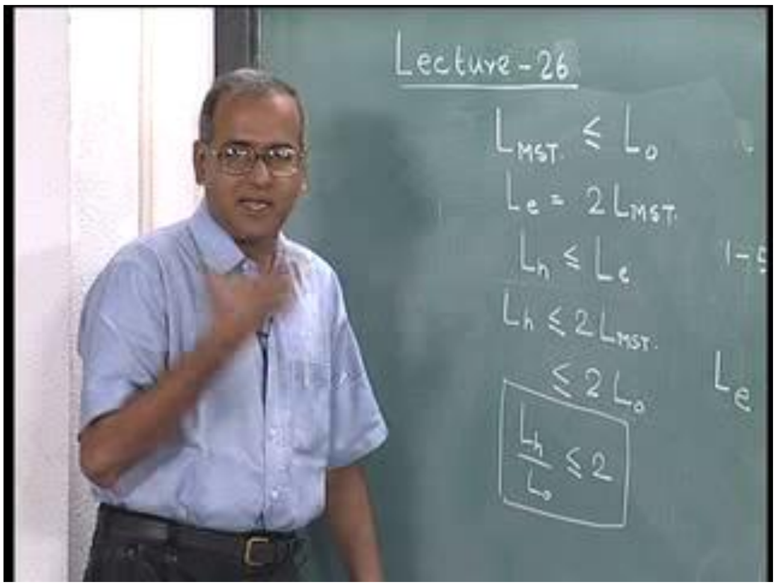
We can do one more thing. We can go from 1 to 5, 5 to 4, 4 to 3, 3 to 2 and 2 to 1. You could do 1 to 5, 5 to 4, 4 to 3, 3 to 2, 2 to 1 and L is 41. 1 to 5 is 7, 5 to 4 is 7 plus 6 13, 4 to 3 13 plus 8 21, 3 to 2 31 plus 10 is 41. Compute again, 1 to 5 is 7, 5 to 4 7 plus 6 13, 4 to 3 13 plus 8 21, 3 to 2 21 plus 10 equal to 31, 2 to 1 plus 10 is 41. Like this, it is possible evaluate as many solutions through this. The only condition being, at the end of it, we should be able to move in this and then come back to 1, we should not move backwards.

We should only keep moving forward and try to get as many solutions as we can. We should not move backwards, for example, we cannot do this. We cannot do for example 1 to 4 or 1 to 2, 2 to 4, 4 to 3, 3 to 5, 5 to 1. We should not do because we would have moved backwards. In this type of a heuristic algorithm, after we get this, you could go back and evaluate as many solutions as you can from this and then take the best. In this example, the best happens to be here, as well as here, which give us the best solution. This is 1 5 2 4 3 1 with L equal to 34. We have got the solution with L equal to 34 here but then, unless we know that it is optimal through some other means, we cannot say that it is optimal.

We would now provide this as a heuristic solution. Of course, we know earlier that this is the optimal solution. So this particular heuristic gives us a solution with L equal to 34. This heuristic is called twice around the tree heuristic and is a very popular heuristic from the literature. It is also easy to understand why it got its name, because in this heuristic, the most important step is to compute the minimum spanning tree and then duplicate it and then get this. It is like moving around the minimum spanning tree twice and therefore, it is called twice around the tree heuristic. We have to find out, can we possibly find out a good analysis of this. How good is this heuristic? We saw an earlier that a goodness of a heuristic is given by this expression called $L_h$ by $L_0$. Now what is the expression for $L_h$ by $L_0$ for this particular heuristic? Let us try and get that expression for this particular heuristic.

The first step of this heuristic was the computation of the minimum spanning tree. So we computed $L_{MST}$. We also said earlier that $L_{MST}$ is a lower bound to $L_0$ because removal of any edge from $L_0$ would give us a spanning tree and $L_{MST}$ is less than or equal to L of any spanning tree. Therefore, $L_{MST}$ is less than or equal to $L_0$. Of course, this is a under the assumption that all $C_{ij}$s are greater than or equal to 0. So under the assumption that all cause are non - negative $L_{MST}$ is less than or equal to $L_0$.

(Refer Slide Time: 42:13)



What we did here was, we found out another $L_e$ which is called the L Eulerian. This Eulerian was obtained by duplicating the edges of the minimum spanning tree. We got the minimum spanning tree here and then we duplicated the edges. We wrote all the 8 edges in a certain order. L Eulerian we already mentioned is 52, which is twice $L_{MST}$ because L Eulerian is a obtained by duplicating the edges of the MST. We have L Eulerian is 2 times $L_{MST}$.

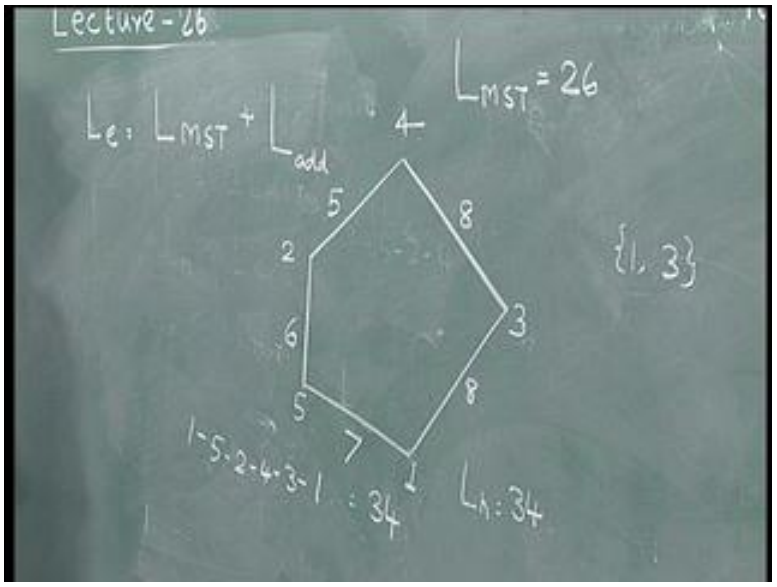
The next thing we did was from this L Eulerian, we picked up several solutions, each of which is a heuristic solution to the TSP. We found out the best out of that. Let us take the best of that, which is this 34 and we called this as $L_h$. But every $L_h$, length of every heuristic, every $L_h$ we will show that this will be less than or equal to $L_e$. If we take this, how did we get this solution? This is 1 5 plus 5 2 2 4 4 3 and 3 1. Now 1 5 is common, 5 2 is common, 2 4 is here, 4 3 is here, but this heuristic has 3 1, whereas $L_e$ has 3 4 plus 4 2 plus 2 5 plus 5 1.

This distance matrix satisfies triangle inequality, which means any $C_{ij}$ is less than or equal to $C_{jk}$ plus $C_{ki}$. If we do that, if the given matrix satisfies triangle inequality, then 3 to 4 plus 4 to 2 is more than 3 to 2. So 3 to 2 plus 2 to 5 is more than 3 to 5 and 3 to 5 plus 5 to 1, is more than 3 to 1. Therefore, 3 to 4 plus 4 to 2 plus 2 to 5 plus 5 to 1 is more than 3 to 1. Therefore, you will see that, $L_e$ is greater than or equal to $L_h$, from which $L_h$ is less than or equal to $L_e$. For every one of these solutions, which we may call as $L_h$, if the matrix satisfies triangle inequality then $L_h$ is less than or equal to $L_e$. Even if we take something like this, if we take 1 to 5, 5 to 4, say before let us take this. We said 1 to 3, 3 to 4, 4 to 2, 2 to 5, 5 to 1. Now 1 to 5 plus 5 to 2 plus 2 to 4 plus 4 to 3 is greater than or equal to 1 to 3 by triangle inequality. Therefore, the $L_h$ corresponding to this will be less than or equal to $L_e$.

If we take this 1 to 5, 5 to 4, 4 to 3, 3 to 2, 2 to 1, 1 to 5, 5 to 2 plus 2 to 4, is greater than or equal to 5 to 4, 4 to 3, 3 to 4 plus 4 to 2, is greater than or equal to 3 to 2 and 2 to 5 plus 5 to 1, is greater than or equal to 2 to 1, Therefore, this $L_h$ will be smaller than $L_e$ or $L_e$ is greater than or equal to this $L_h$. So $L_e$ is greater than or equal to every one of these that is here. Therefore, $L_e$ is greater than or equal to the best of these, which we call as $L_h$.

Since $L_e$ is greater than or equal to every one of these, it has to be greater than or equal to the best of these. $L_e$ is greater than or equal to $L_h$ or $L_h$ is less than or equal to $L_e$. Now $L_h$ is equal 2 times $L_{MST}$. This is less than or equal to 2 times $L_{MST}$ now, from this result $L_{MST}$ is less than or equal to $L_0$, so $L_h$ is less than or equal to 2 $L_0$. Now from this we can derive $L_h$ by $L_0$ is less than or equal to 2, which is an important result about the performance of the twice around the tree heuristic, This heuristic, we can say with confidence that this heuristic will guarantee a solution, always such that the solution $L_h$ by $L_0$ is less than or equal to 2. No matter what the problem size is, we will get $L_h$ by $L_0$ less than or equal to 2, provided all the $C_{ij}$s are greater than or equal to 0. This satisfies triangle inequality.

Under these 2 conditions, we can say that twice around the tree heuristics, no matter what the size of the problem is, no matter what the numbers are, as long as those numbers are real numbers and they satisfy triangle inequality. We can say that $L_h$ by $L_0$t is less than or equal to 2. This is some form of an analysis of a heuristics and a construction heuristic because, we have constructed the heuristic solution from this, so this is how the twice around the tree heuristic works. It is called twice around the tree because we go through the minimum spanning tree twice. It also has a performance measure of 2. We look at a third another heuristic which is also based on the minimum spanning tree.
(Refer Slide Time: 48:18)



We have already seen that this is a minimum spanning tree. So 1 to 3 2, 3 to 4, 4 to 2, 2 to 5 and 5 to 1. Here, we have duplicated it so I am just removing this. We say this is the minimum spanning tree with values 3 to 4 is 8, 2 to 4 is 5, 2 to 5 is 6 and 1 to 5 is 7 with $L_{MST}$ equal to 26, 8 plus 5 equal to 13 plus 6 equal to 19 plus 7 equal to 26. Let us look at the minimum spanning tree and let us look at some properties of it. This is a graph which is a collection of nodes and arcs. Now with any graph, there is an interesting property that the number of vertices with an odd degree is even. Let us explain this now. What is a degree of a vertex? The degree of a vertex is a number of edges that are incident onto the vertex. This has a degree 1 because there is only one vertex, this has degree 2, this has degree 2, this has degree 2 and this has degree 1. Since every edge is counted twice, because every edge connects 2 vertices. It contributes the degree of 2 vertices. So the total degree of any graph is an even number because it is multiplied by 2 times the numbers of edges.

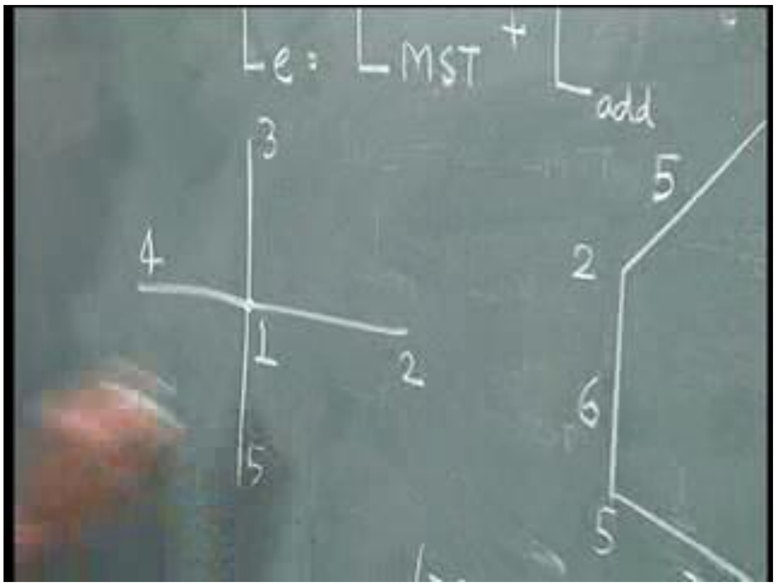
The total degree being even, it is obvious that the number of vertices with odd degree will have to be even. In this example, there are with four edges in this, so sum of the degree has to be eight. We will see 1 plus 2 plus 2 plus 2 plus 1. We got 1 plus 2 plus 2 plus 2 plus 1. This is odd degree; this is odd degree, so number of vertices that have odd degree will have to be even. Let us do with any graph and therefore, it is true with a minimum spanning tree, as well, now find out the number of vertices that have an odd degree. In this case it is just 1 and 3, there are 2 vertices that have odd degree, which happen to be 1 and 3.

If you have only 2 vertices that have odd degree, then just go back and join the vertices. Just join the vertices and you add another 8. We said if you have only 2 vertices that have odd degree, the only thing you will do is, you will simply join these 2 and when you simply join these 2, you will get one solution, which is a circuit. This is because you have simply added one edge into a minimum spanning tree. If you have only 2 vertices that have odd degree, then you will simply put this into the spanning tree.

You also realize that by adding such a vertex, you get one more Eulerian, which means you can start from 1. 1 to 5, 5 to 2, 2 to 4, 4 to 3, 3 to 1. You go through every edge once and only once and you come back to the starting point. It is also Hamiltonian, that from every vertex, you can go to every other vertex and come back to the starting point. In this case, the L Eulerian will be equal to $L_{MST}$ plus sum L that we have added. I am just going to call it $L_{add}$ and because the number of vertices having odd degree is only 2. We just end up adding only one edge into this solution. Therefore, this solution is 1 to 5, 5 to 2. 2 to 4, 4 to 3, 3 to 1, there is only one solution that you get here, if you have 2 vertices coming here,

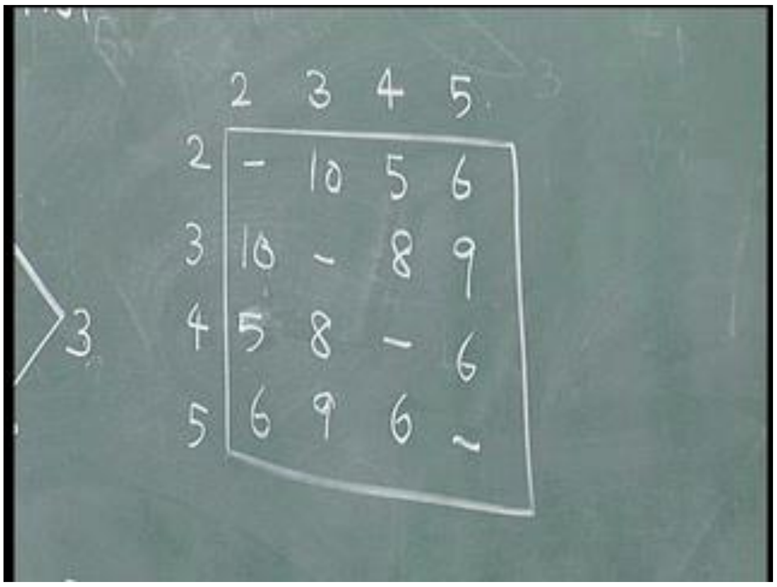
You get only one solution. Now the length of this solution is 1 to 5 is 7, 5 to 2 7 plus 6 equal to 13, 2 to 4 13 plus 5 18, 4 to 3 18 plus 8 26, 3 to 1 is 34. There is only one solution which we give as a heuristic solution. $L_h$ equal to 34, now we have provided a heuristic solution to this problem, whose length equal to 34. This algorithm is called heuristics based on matching or let me explain the matching as well. Let me explain the matching differently. We have seen only one case, where the number of vertices with odd degree happens to be 2. When the number of vertices with odd degree happens to be 2, we simply add that particular edge into the spanning tree to get a feasible solution to the Travelling Salesman Problem which becomes your $L_h$ but what happens if you have more than 2.

(Refer Slide Time: 54:16)



For example, if the minimum spanning tree for a 5 - city problem is like this, let us say with 1 here, let me say 2 3 4 and 5. This is a case where the number of vertices with odd degree is 4, as against number of vertices with odd degree equal to 2. In this case, you have to go back and create the vertices with the odd degree or 2 3 4 and 5.

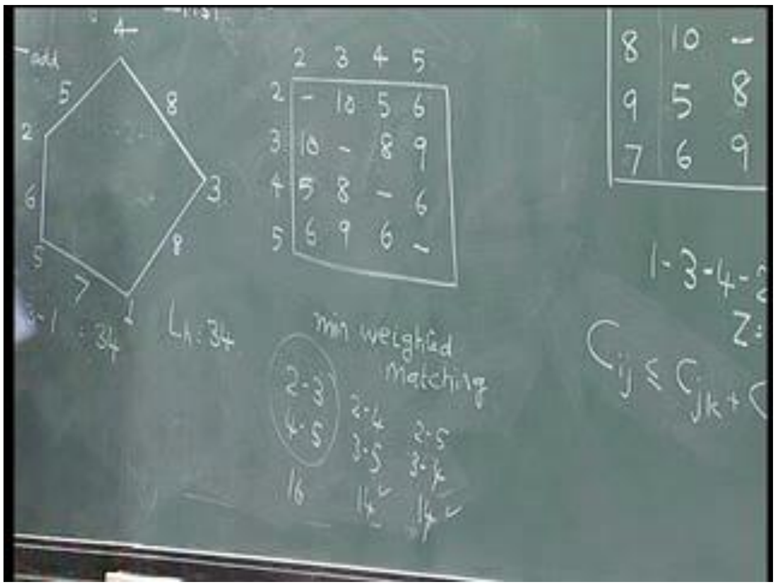
(Refer Slide Time: 54:50)



$$\begin{array}{c|cccc|} & 2 & 3 & 4 & 5 \\ \hline 2 & - & 10 & 5 & 6 \\ 3 & 10 & - & 8 & 9 \\ 4 & 5 & 8 & - & 6 \\ 5 & 6 & 9 & 6 & - \\ \hline \end{array}$$

You will create another matrix with 2 3 4 and 5. 2 3 4 and 5, this is assuming that 2 3 4 and 5 are the vertices with odd degree. Create this matrix. We would get a matrix which is like this. Dash 10 5 6, we would get this portion of the matrix. 10 dash 8 9 5 8 dash 6 9 6 dash.

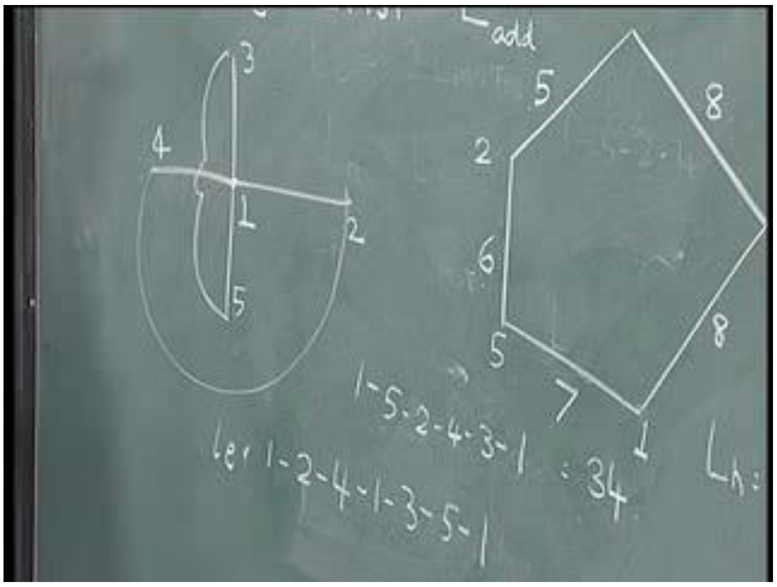
(Refer Slide Time: 55:43)



We would get a matrix like this. What we have done from this matrix is to get a minimum weighted matching. What is this minimum weighted matching? There are 4 nodes 2 3 4 and 5. For example, if I take 2 to 3 and 4 to 5, it is called a matching. Because there are 4 nodes, I have created 2 edges, such that the nodes are not common but all the vertices are included. If I look at 2 3 4 and 5, I may have 2 3 4 5. I may have 2 4 3 5 and I may have 2 5 3 4.

Because there are 4 vertices, there can be only 2 matching, each matching has one edge and 2 nodes. If we take this particular matching, this matching comprises of all these 4, 2 edges but all 4 are kept here. So 3 such matching are possible and the weights of these are 2 3 4 5 is 16 2 4 is 5, 3 5 is 5 plus 9 equal to 14, 2 5 3 4 is also 14. Out of the three possibilities that we have here, the minimum one is either this or this so we could choose any one of them, choose the minimum weighted matching and then add it onto this.

(Refer Slide Time: 57:28)



Suppose we take 2 4 3 5, then we end up doing 2 4 3 5 onto this. Then we will get a solution 1 to 2, 2 to 4, 4 to 1, 1 to 3, 3 to 5, 5 to 1. This will be the L Eulerian if you super impose this. We continue the discussion on this in the next lecture.