Practice 15. Branch and bound for the Travel Salesman Problem

Mayra Cristina Berrones Reyes

May 7, 2020

1 Introduction

The traveling salesman problem (TSP) consists of a set of nodes that represent "cities", and and joining them there are edges with a cost or "distance" that we need to pay to visit all the cities. In this case, we are trying to find the shortest or cheapest way of "visiting all the cities". The ordering is called tour or circuit.

This type of problem is similar to the Hamiltonian cycle. Thanks to a previous practice [1] we can say that the TSP problem is then $\mathcal{NP}-$ complete.

This problem is very old, and can be known by different names. There are also several ways to solve this problem, but in this practice we are going to be solving it by the branch and bound method. This consists in a exhaustive search for the best solution in a set. Each branching step reduces the search space, in order to make it easier than the original [Applegate et al., 2006].

2 Solving TSP with branch and bound

In this section we are getting an example with four nodes. We can see the distance matrix in Equation 1. With a code in Python [2] we generate a random matrix, and then we solve it step by step using the branch and bound method, in specific, the best bound first search strategy which rules to the subset with the smaller lower bound [Smith, 1979].

[1] https://github.com/mayraberrones94/Analisis_Algoritmos/blob/master/Practica_5/Practica5.pdf

[2] https://github.com/mayraberrones94/Analisis_Algoritmos/blob/master/Practica_15/tsp.py

$$\begin{bmatrix} 0 & 31 & 44 & 12 \\ 6 & 0 & 46 & 44 \\ 26 & 4 & 0 & 38 \\ 24 & 16 & 10 & 0 \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} \infty & 31 & 44 & 12 \\ 6 & \infty & 46 & 44 \\ 26 & 4 & \infty & 38 \\ 24 & 16 & 10 & \infty \end{bmatrix} \begin{matrix} 12 \\ 6 \\ 4 \\ 10 \end{matrix} = 32$$

In the above matrix we can see that we changed the 0 to ∞, because in this case, the 0 mean something different. The first step is to see which number is the smallest per row. In this matrix they are colored red. The sum of all this numbers is the value of our first reduced matrix. Next we subtract the smaller number to all the elements of the row, and we end up with a matrix like the one seen below, and we will call it our reduced matrix.

$$\begin{bmatrix} \infty & 19 & 32 & 0 \\ 0 & \infty & 40 & 38 \\ 22 & 0 & \infty & 34 \\ 14 & 6 & 0 & \infty \end{bmatrix}$$

To begin our search, first we name node 1 as our root. The reduced cost (r) of this node is going to be the sum we made in the reduced matrix, in this case is 32. Now we take the reduced matrix and begin our search with all the nodes that are adjacent to the node 1. The rules to this are:

1. We turn all the row in our reduced matrix of our parent node to ∞.
2. Then we also turn all our column of the child node we are going to explore to ∞
3. If the node we are exploring is not the last node, we also turn to ∞ the intersection number of the explored node and the root node.
4. We search first by row, then by column for the smaller number. If its zero the row/column stays the same. If not, we subtract it form the row/column and we make a sum of all of this elements.
5. The sum of all this smaller numbers in rows and columns will take the name of r_i.

The formula we are going to be using to get the cost of each node is in Equation 2:

$$c(nodeP, nodeC) + r + r_i = \text{Cost of the node} \tag{2}$$

So now that we know the rules, we begin to explore the nodes that are adjacent to the node parent. We begin with $c(1,2)$:

$$\begin{bmatrix} \infty & 19 & 32 & 0 \\ 0 & \infty & 40 & 38 \\ 22 & 0 & \infty & 34 \\ 14 & 6 & 0 & \infty \end{bmatrix} \rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 40 & 38 \\ 22 & \infty & \infty & 34 \\ 14 & \infty & 0 & \infty \end{bmatrix} \begin{matrix} 0 \\ 38 \\ 22 \\ 0 \end{matrix} = 60 \rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 2 & 0 \\ 0 & \infty & \infty & 12 \\ 14 & \infty & 0 & \infty \end{bmatrix}$$

$$c(1,2) + r + r_i = 19 + 32 + 60 = 111 \tag{3}$$

The result for the first exploration is a cost of 111. So now we move to the right to the next node $c(1,3)$

$$\begin{bmatrix} \infty & 19 & 32 & 0 \\ 0 & \infty & 40 & 38 \\ 22 & 0 & \infty & 34 \\ 14 & 6 & 0 & \infty \end{bmatrix} \rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 38 \\ \infty & 0 & \infty & 34 \\ 14 & 6 & \infty & \infty \end{bmatrix} \begin{matrix} 0 \\ 0 \\ 0 \\ 6 \end{matrix} \rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 38 \\ \infty & 0 & \infty & 34 \\ 8 & 0 & \infty & \infty \end{bmatrix} \rightarrow$$

$$\begin{bmatrix} 0 & 0 & 0 & 34 \end{bmatrix} \rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & 2 \\ \infty & 0 & \infty & 0 \\ 8 & 0 & \infty & \infty \end{bmatrix} \rightarrow 6 + 34 = 40$$

$$c(1,3) + r + r_i = 32 + 32 + 40 = 104 \tag{4}$$

For the next exploration we have a cost of 104, so lastly we move to the right again to the last node adjacent to 1, $c(1,4)$.

$$\begin{bmatrix} \infty & 19 & 32 & 0 \\ 0 & \infty & 40 & 38 \\ 22 & 0 & \infty & 34 \\ 14 & 6 & 0 & \infty \end{bmatrix} \rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 40 & \infty \\ 22 & 0 & \infty & \infty \\ \infty & 6 & 0 & \infty \end{bmatrix} \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix} \rightarrow = 0$$

$$c(1,4) + r + r_i = 0 + 32 + 0 = 32 \tag{5}$$

Finally, in this last exploration we can see the best cost to the adjacent nodes of one, which is 32 and is the relation of $c(1,4)$. The strategy of best bound first search is depicted in Figure 1, where we deprecate the other nodes adjacent to node 1, and we continue to explore the node with the lower bound of cost.

Now that we have to explore the adjacent nodes of the node 4, our reduced matrix changes to the one we made for the exploration of $c(1,4)$. The remaining nodes we have to explore are node 2 and 3. We begin with $c(4,2)$.

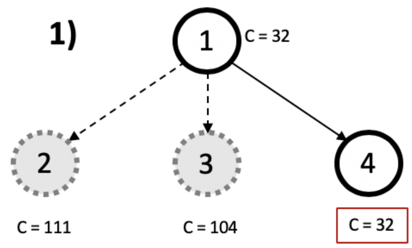

Figure 1: Exploration of nodes adjacent to the root node.

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 40 & \infty \\ 22 & 0 & \infty & \infty \\ \infty & 6 & 0 & \infty \end{bmatrix} \rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 40 & \infty \\ 22 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \end{bmatrix} \begin{matrix} 0 \\ 40 \\ 20 \\ 0 \end{matrix} \rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ \infty & \infty & 0 & \infty \\ 0 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty \end{bmatrix} = 60$$

$$c(4,2) + r + r_i = 6 + 32 + 60 = 98 \tag{6}$$

The cost for this exploration is 98. So now we move to the next node, $c(4,3)$.

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & 40 & \infty \\ 22 & 0 & \infty & \infty \\ \infty & 6 & 0 & \infty \end{bmatrix} \rightarrow \begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty \end{bmatrix} \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix} = 0$$

$$c(4,3) + r + r_i = 0 + 32 + 0 = 32 \tag{7}$$

Now we find that the cost of the exploration of $c(4,3)$ is the best lower bound, so we take this node, and repeat the same steps we did when we changed parent node. We take the reduced matrix of node 3, and explore the last node we have available, $c(3,2)$.

$$\begin{bmatrix} \infty & \infty & \infty & \infty \\ 0 & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \\ \infty & \infty & \infty & \infty \end{bmatrix} \rightarrow \begin{matrix} 0 \\ 0 \\ 0 \\ 0 \end{matrix} = 0$$

$$c(3,2) + r + r_i = 0 + 32 + 0 = 32 \tag{8}$$

We easily find the lower bound here, because most of our matrix is already filled with ∞. In Figure 2 we see the rest of the exploration, where we deprecate the node 2 when node 4 was the parent, and the final path taken with this

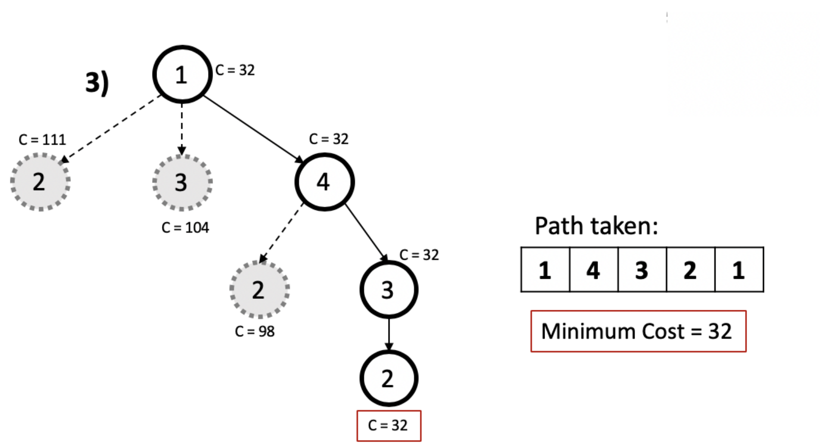

Figure 2: Diagram of divide and conquer steps.

algorithm. At the end we find that the lowest bound was 32.

Finally in Figure 3 we have the output of the terminal, and we can see that the results match.

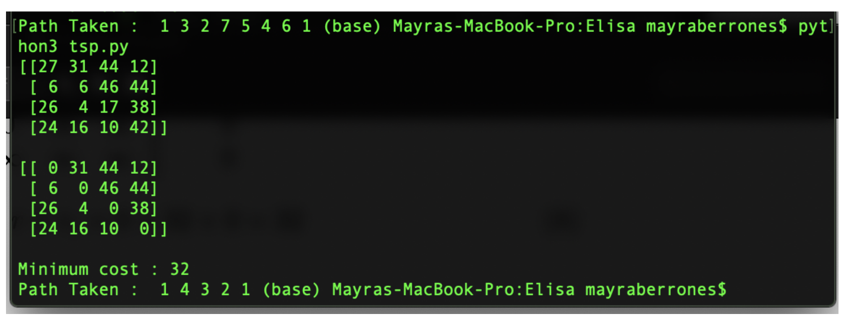

Figure 3: Terminal with the results of the branch and bound of the TSP.

3 Conclusions

For the time complexity I could not find much. We know that TSP is $\mathcal{NP}-$ complete, but one of the books [Smith, 1979] mentions that the complexity depends mostly on the size of the problem so I made a little experiment with diferent size of matrix, starting with the experiment shown in this practice of four nodes, up to 25 nodes. We can see in Figure 4 that the time grows exponentially around the 17 and 20 node, so we can confirm that this type of problem is feasible to solve this way on small instances.

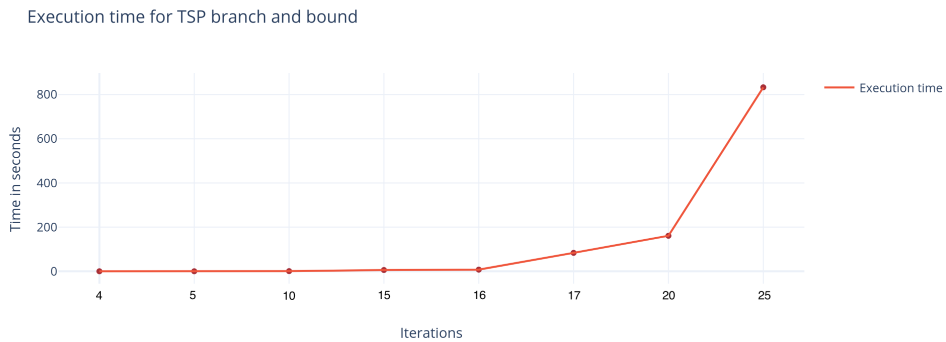

Figure 4: Execution time for branch and bound for TSP.

References

[Applegate et al., 2006] Applegate, D. L., Bixby, R. E., Chvatal, V., and Cook, W. J. (2006). *The traveling salesman problem: a computational study*. Princeton university press.

[Smith, 1979] Smith, D. R. (1979). *On the computational complexity of branch and bound search strategies*. Naval post graduate school, Monterrey California.