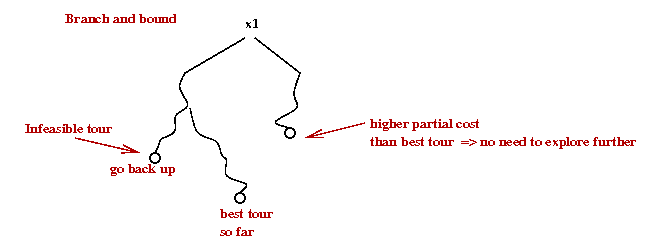
# Exact Solution Traveling Salesman Analysis: “*Branch And Bound”*

For determining the exact solution, the easiest approach to consider would be that of the “Naïve Approach” or “Brute Force Method” in which case every permutation is examined. A more efficient approach (amongst many) to determining the exact solution however involves either applying a *Branch and Bound* method.

The underlying idea behind the B&B method is that an optimal solution is found by forming a rooted tree of candidate solutions starting at the route node (starting city). From this point, each branch is checked against the upper and lower estimated bounds of the optimal solution, and the branch is discarded if it cannot produce a better solution than the best one found so far. [1]

Below is a simple graphical representation of the B&B method.

[2]

To determine this upper and lower brands, we first need to create an adjacency matrix for the graph in question. After which the matrix is reduced, and we obtain a cost of reduction. For each subsequent node we assign a “cost” value in the following manner:  
(The following matrixes are a direct quote/example taken from: <https://people.eecs.berkeley.edu/~demmel/cs267/assignment4.html>)

i\j 1 2 3 4 5 6 7

\ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 |Inf 3 93 13 33 9 57

2 | 4 Inf 77 42 21 16 34

3 | 45 17 Inf 36 16 28 25

4 | 39 90 80 Inf 56 7 91

5 | 28 46 88 33 Inf 25 57

6 | 3 88 18 46 92 Inf 7

7 | 44 26 33 27 84 39 Inf

[3]

* Reduce this new matrix to obtain an additional cost of reduction

i\j 1 2 3 4 5 6 7

\ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Cost of reduction:

1 |Inf 0 83 9 30 6 50 3

2 | 0 Inf 66 37 17 12 26 4

3 | 29 1 Inf 19 0 12 5 16

4 | 32 83 66 Inf 49 0 80 7

5 | 3 21 56 7 Inf 0 28 25

6 | 0 85 8 42 89 Inf 0 3

7 | 18 0 0 0 58 13 Inf 26

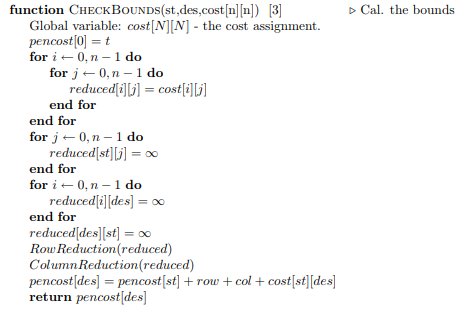
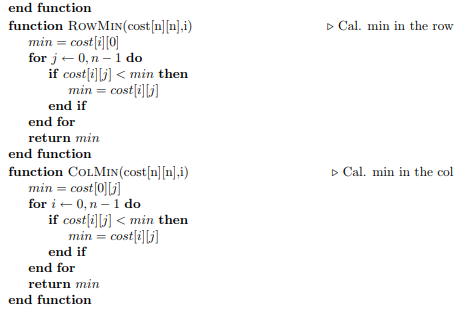
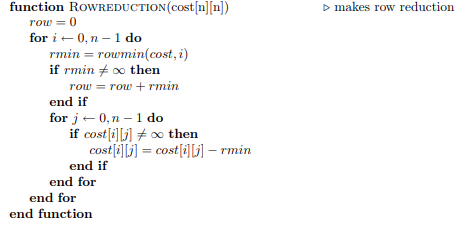
Total Cost of Reduction: 84

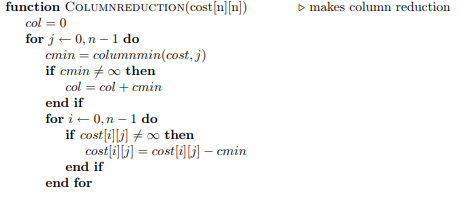
* Add the matrix reduction costs to the path cost C(Root Node, Node-k) + R+ Rk’  
  (Where C() is the distance between two nodes) R is the Root Cost of Reduction, and Rk’ is the cost of reduction of subsequently analyzed node k where subsequent matrix reductions are done per row AND per column.
* Repeat procedure for all connected nodes (obtain the cost of the children nodes, distance + R + R’)
* Select the node with the lowest cost
* Repeat process for that node

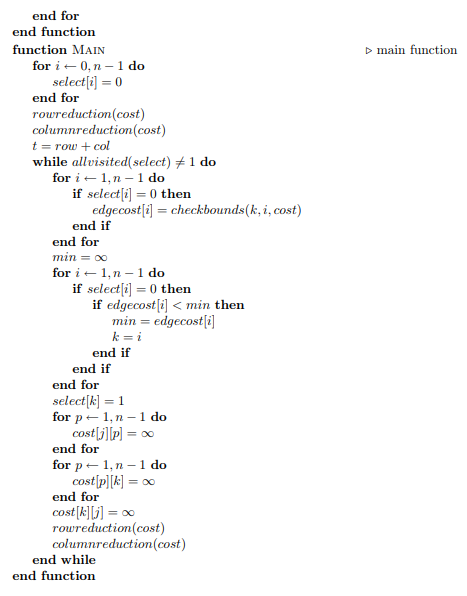
Process is repeated until we have visited all nodes (cities) and thus the optimal TSP path has been found.

For a much more detailed explanation of the outlined procedure please visit <https://www.youtube.com/watch?v=1FEP_sNb62k> and/or <https://people.eecs.berkeley.edu/~demmel/cs267/assignment4.html>

Pseudocode: (The following pseudo code is a direct quote from <http://cs.indstate.edu/cpothineni/alg.pdf> ) [4]





# Bibliography

|  |  |
| --- | --- |
| [1] | "Branch and nound," [Online]. Available: https://en.wikipedia.org/wiki/Branch\_and\_bound. [Accessed 30 May 2018]. |
| [2] | "The Traveling Salesman Problem (TSP)," [Online]. Available: https://www2.seas.gwu.edu/~simhaweb/champalg/tsp/tsp.html. [Accessed 30 May 2018]. |
| [3] | "CS267. Assignment 4: Traveling Salesman Problem," 1 April 1996. [Online]. Available: https://people.eecs.berkeley.edu/~demmel/cs267/assignment4.html. [Accessed 5 May 2018]. |
| [4] | C. Pothineni, "http://cs.indstate.edu/cpothineni/," 13 December 2013. [Online]. Available: http://cs.indstate.edu/cpothineni/alg.pdf. [Accessed 30 May 2018]. |
| [5] | "CS267. Assignment 4: Traveling Salesman Problem," 1 April 1996. [Online]. Available: https://people.eecs.berkeley.edu/~demmel/cs267/assignment4.html. [Accessed 5 May 2018]. |