

Travelling Salesman Problem

by Hung Nguyen

Problem statement

“Given a list of cities and the distances between each pair of cities, find the shortest possible route that visits each city and returns to the origin city”

- Graph problem
- Find the shortest path that visits all the nodes and come back to the origin node.

Applications

- Vehicle routing
 - UPS delivery
 - School bus routing
- Let's go micro...
 - Computing Wiring
 - PCB production

5,704 results for *travelling salesman problem*

travelling salesman problem search advanced

Books, media & more 514 results

1. [The traveling salesman problem: a computational study \(2006\) Book](#) 

2. [The Traveling Salesman Problem: a Computational Study \(2011\) Book \(Online\)](#)
Applegate, David L. 

3. [The Traveling salesman problem: a guided tour of combinatorial optimization \(1985\) Book](#) 

4. [The traveling salesman problem and its variations \(2007\) Book \(Online\)](#) 

5. [The traveling salesman problem and its variations \(2002\) Book \(Online\)](#)

6. [Routing methods: principles for handling multiple traveling salesman problems \(1965\) Book](#) 

Articles & more 5,188 results

1. [An approximation algorithm for a special case of the asymmetric travelling salesman problem. \(Academic Journal\)](#)

2. [The multi-stripe travelling salesman problem \(Academic Journal\)](#)

3. [Discrete symbiotic organisms search algorithm for travelling salesman problem \(Academic Journal\)](#)

4. [A linear time algorithm for the 3-neighbour Travelling Salesman Problem on a Halin graph and extensions \(Academic Journal\)](#)

5. [Variation of ant colony optimization parameters for solving the travelling salesman problem \(Conference\)](#)

6. [Fast heuristic algorithm for travelling salesman problem \(Conference\)](#)

7. [A node current-based 2-index formulation](#)

Questions about y

Exact algorithm

Brute force...

Time complexity: $O(n!)$

Try all paths – $n!$ permutation of edges.

Not practical!

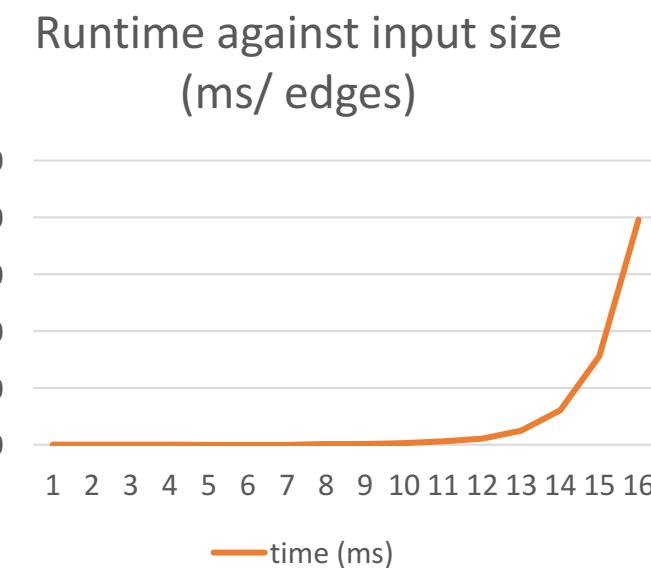
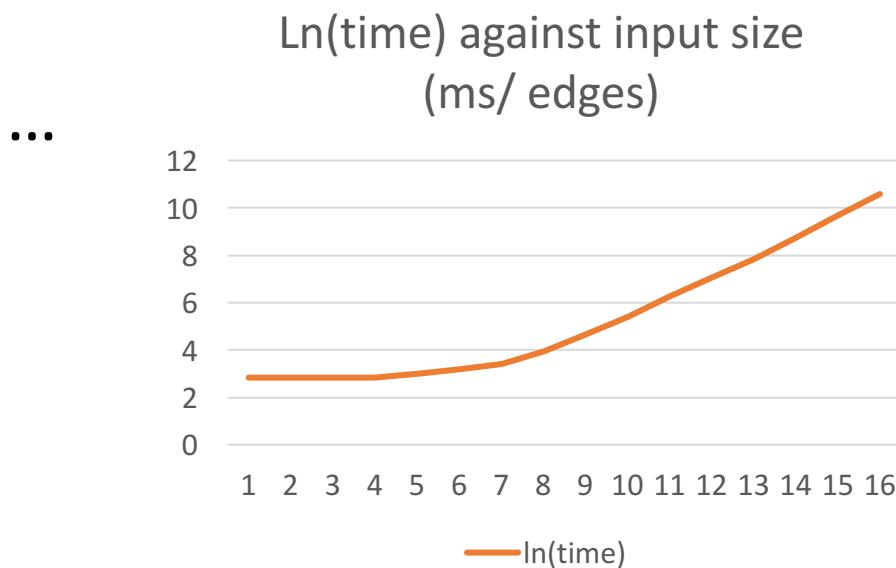
Exact algorithm (cont.)

DP solution: Held-Karp algorithm

Time complexity: $O(2^n * n^2)$

Implementation: <https://github.com/CarlEkerot/held-karp>

Also, not practical!

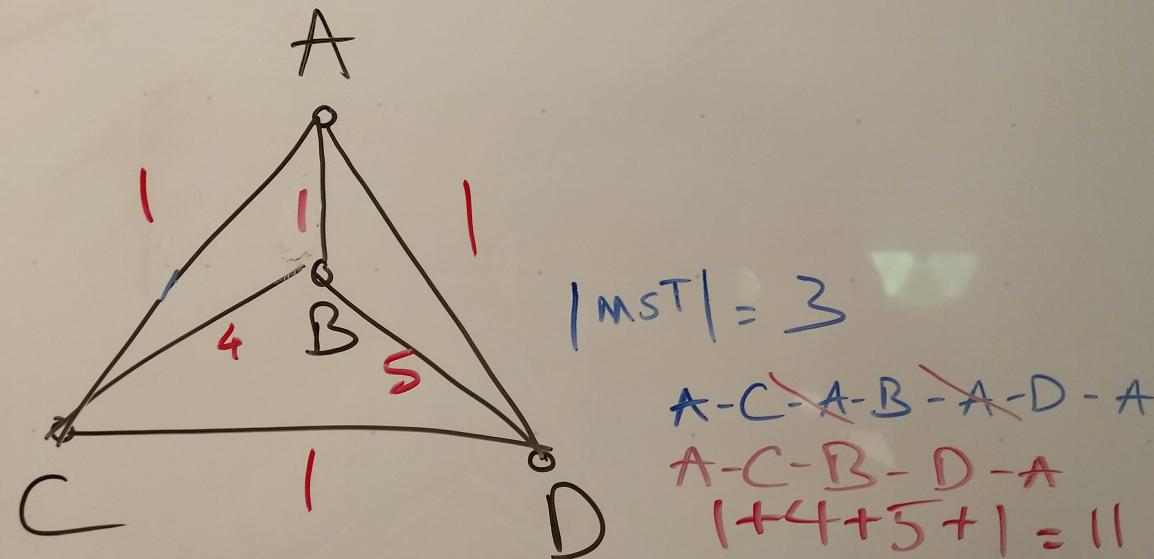


Heuristic – Approximation

- Idea: find some thing “cheaper” but “good enough”.
 - Walk on the MST edges to complete the cycle (Here we duplicate the nodes, the result is Hamiltonian cycle)
 - We “collapse” nodes on the path
- How good is the solution?
 - Minimum Spanning Tree: the cost of the optimal path cannot be smaller than the size of MST (by MST definition) $C \leq 2 \times MST \leq 2 OPT$
 - 2-approximation: we can guarantee the found path is at most twice as long as the optimal path.
- Time complexity: $O(n^2 \log n)$
- Let’s try out some code!
- There is a caveat: only works with metric TSP (triangle inequality holds)
 - Why?

Triangle inequality

White board....



The upper-bound does not hold on cost
i.e. $C \leq 2 \times MST$

The reason is when we reduce the path, the cost might increase, i.e. $\text{Path}(A - C - B) < \text{Path}(A - B)$

Sources

- MST: Code Archive
- Fleury: <https://www.geeksforgeeks.org/fleurys-algorithm-for-printing-eulerian-path/>
- Held-karp: <https://github.com/CarlEkerot/held-karp>
- This presentation: <https://github.com/Hank-TNguyen/CMPUT-403-demo>

Thx!

UVa problem

- https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=52
 - UVa_116 – Unidirectional TSP