

Theoretical Analysis of the Double-Tree Algorithm for TSP

1 Introduction

The Traveling Salesman Problem (TSP) is a fundamental NP-hard optimization problem in computer science. Given a complete graph $G = (V, E)$ with n vertices and non-negative edge costs satisfying the triangle inequality (metric TSP), the objective is to find a Hamiltonian cycle of minimum total cost. The double-tree algorithm (also called MST-based algorithm) provides a polynomial-time 2-approximation for metric TSP.

2 Algorithm Description

The algorithm proceeds in three phases:

Algorithm 1 Double-Tree Algorithm for Metric TSP

Require: Complete graph $G = (V, E)$, cost function $c : E \rightarrow \mathbb{R}^+$ satisfying triangle inequality

Ensure: Hamiltonian cycle H

- 1: Construct a minimum spanning tree T of G
 - 2: Duplicate each edge in T to create Eulerian multigraph M
 - 3: Find an Eulerian tour E in M
 - 4: Convert E to Hamiltonian cycle H by shortcutting repeated vertices
 - 5: **return** H
-

3 Theoretical Analysis

3.1 Correctness

The algorithm produces a valid Hamiltonian cycle because:

1. The MST T connects all vertices (by definition of spanning tree)
2. Doubling edges makes all vertices have even degree, guaranteeing existence of Eulerian tour
3. Shortcutting preserves connectivity and visits each vertex exactly once due to triangle inequality

3.2 Approximation Ratio

The double-tree algorithm produces a Hamiltonian cycle with cost at most $2 \cdot OPT$, where OPT is the optimal TSP tour cost.

Proof. Let T be the minimum spanning tree, H be the output Hamiltonian cycle, and E be the Eulerian tour.

1. **MST bound:** Removing any edge from an optimal TSP tour yields a spanning tree. Thus:

$$c(T) \leq OPT \quad (1)$$

2. **Eulerian tour cost:** Since we double every edge of T :

$$c(E) = 2 \cdot c(T) \leq 2 \cdot OPT \quad (2)$$

3. **Shortcutting bound:** The triangle inequality ensures that for any vertices u, v, w :

$$c(u, w) \leq c(u, v) + c(v, w) \quad (3)$$

Therefore, shortcutting (replacing a path $u \rightarrow v \rightarrow w$ with direct edge $u \rightarrow w$) never increases the cost:

$$c(H) \leq c(E) \quad (4)$$

4. **Combining inequalities:**

$$c(H) \leq c(E) = 2 \cdot c(T) \leq 2 \cdot OPT \quad (5)$$

Thus the algorithm achieves a 2-approximation ratio. \square

3.3 Time Complexity Analysis

The double-tree algorithm runs in $O(n^2)$ time for n vertices with an adjacency matrix representation.

Proof. We analyze each component of the algorithm:

1. **MST Construction (Prim's Algorithm):**

- Initialization: $O(n)$
- Main loop executes n times
- Finding minimum key: $O(n)$ per iteration, total $O(n^2)$
- Updating keys: Each vertex's neighbors checked once, $O(n)$ per iteration, total $O(n^2)$
- **Total:** $O(n^2)$

2. **Eulerian Tour Construction:**

- Creating adjacency lists from MST: $O(n)$ (MST has $n - 1$ edges)
- Hierholzer's algorithm for Eulerian tour:

- Each edge visited exactly once: $O(n)$ edges in doubled MST
- Stack operations: $O(1)$ per edge
- **Total:** $O(n)$

3. Shortcutting:

- Processing Eulerian tour of length $O(n)$
- Checking visited vertices: $O(1)$ amortized using hash set
- **Total:** $O(n)$

The total time complexity is dominated by MST construction:

$$T(n) = O(n^2) + O(n) + O(n) = O(n^2) \quad (6)$$

For sparse graphs with adjacency list and binary heap, Prim's algorithm runs in $O(m \log n)$ where m is the number of edges. However, for complete graphs in metric TSP, $m = \Theta(n^2)$, making $O(n^2)$ optimal for this representation. \square

3.4 Tightness of Approximation Ratio

The approximation ratio of 2 is tight for the double-tree algorithm.

Proof. Consider n points equally spaced on a circle of radius 1:

- Optimal TSP tour: Follow the circumference, cost $OPT = n \cdot 2 \sin(\pi/n) \approx 2\pi$
- MST: A star connecting all points to center, cost $c(T) = n$
- Eulerian tour: Traverse each edge twice, cost $2n$
- After shortcutting: Algorithm may produce tour visiting points in order $0, 2, 4, \dots, 1, 3, 5, \dots$ with cost approaching $2n$ as $n \rightarrow \infty$

The approximation ratio approaches:

$$\frac{c(H)}{OPT} \approx \frac{2n}{2\pi} = \frac{n}{\pi} \rightarrow 2 \text{ as } n \rightarrow \infty \quad (7)$$

Thus the bound is asymptotically tight. \square

4 Comparison with Christofides Algorithm

Algorithm	Approximation Ratio	Time Complexity	Key Idea
Double-Tree	2	$O(n^2)$	MST + Edge Doubling
Christofides	1.5	$O(n^3)$	MST + Perfect Matching

Table 1: Comparison of TSP approximation algorithms

5 Implementation Details

The C++ implementation uses:

- Prim’s algorithm for MST construction
- Adjacency list representation for MST
- Hierholzer’s algorithm for Eulerian tour
- Hash set for efficient shortcutting
- Euclidean distance metric

6 Conclusion

The double-tree algorithm provides a simple, efficient 2-approximation for metric TSP with proven theoretical guarantees. While Christofides’ algorithm achieves a better approximation ratio (1.5), the double-tree algorithm remains valuable due to its simplicity and lower constant factors. The algorithm demonstrates fundamental connections between spanning trees, Eulerian graphs, and Hamiltonian cycles in approximation algorithm design.