1 NP-Hardness of Finding the Optimal Order

1.1 Problem Definition

Definition 1 (Finding Optimal Order Problem)**.** *Given a graph $G = (V, E)$ with $|V| = n$ vertices, the Finding Optimal Order (FOO) problem seeks a vertex traversal order $\pi = (v_1, v_2, \ldots, v_n)$ that minimizes the total cost:*

$$Cost(\pi) = \sum_{i=1}^{n} C_i$$

subject to the following conditions:

(i) Each vertex is visited exactly once

(ii) C_i denotes the cost of selecting vertex v_i at position i to perform a round of pruned Breadth-First-Search (BFS) from this vertex [1], which depends on the previous selections $(v_1, \ldots, v_{i-1})$

(iii) The costs exhibit a general decreasing trend, i.e., $C_i < C_{i-1}$ for most (but not necessarily all) consecutive pairs because of the pruning mechanism in Pruned Landmark Labeling (PLL) [1]

(iv) The exact cost function is unknown a priori and must be evaluated for each specific order

1.2 Main Result

Theorem 2. *The Finding Optimal Order (FOO) problem is NP-hard.*

Proof. We establish the NP-hardness of FOO via a polynomial-time reduction from the Traveling Salesman Problem (TSP).

1.2.1 TSP Instance

Consider a TSP instance consisting of a complete weighted graph $G_{\text{TSP}} = (V, E, w)$, where $w : E \to \mathbb{R}^+$ assigns positive weights to edges. The objective is to find a Hamiltonian cycle of minimum total weight.

1.2.2 Reduction Construction

Given a TSP instance, we construct a corresponding FOO instance through the following steps:

1. **Vertex Set**: Retain the same vertex set V with $|V| = n$.

2. **Parameter Definitions**:

   - Let $W_{\max} = \max_{(u,v)\in E} w(u, v)$ denote the maximum edge weight

- Define $M = n \cdot W_{\max}$

3. **Cost Function (with wrap-around):** We take indices modulo n, so $v_{n+1} = v_1$. Define

$$C_i = M - (i-1)\,W_{\max} + w(v_i, v_{i+1}) \quad \text{for } i = 1, \ldots, n. \tag{1}$$

1.2.3 Properties of the Construction

The constructed FOO instance satisfies the problem requirements:

1. **Trend remark (informal):** The term $-(i-1)W_{\max}$ induces a decreasing drift across C_i, while the edge terms $w(v_i, v_{i+1})$ may fluctuate within $[0, W_{\max}]$. We do not rely on monotonicity in the proof.

2. **Non-strict Monotonicity**: When $w(v_{i-1}, v_i) = W_{\max}$, we obtain $C_i = C_{i-1}$, confirming that the decrease is not strictly monotonic.

3. **Dependency on Previous Selections**: The cost C_i explicitly depends on the previously selected vertex v_{i-1} through the edge weight term $w(v_{i-1}, v_i)$.

1.2.4 Cost Equivalence

For any vertex order $\pi = (v_1, \ldots, v_n)$,

$$\begin{aligned}
\mathrm{Cost}_{\mathrm{FOO}}(\pi) &= \sum_{i=1}^{n} \big[M - (i-1)\,W_{\max} + w(v_i, v_{i+1})\big] && (2)\\
&= nM \;-\; W_{\max} \sum_{i=1}^{n} (i-1) \;+\; \sum_{i=1}^{n} w(v_i, v_{i+1}) && (3)\\
&= nM \;-\; W_{\max}\, \frac{n(n-1)}{2} \;+\; \sum_{i=1}^{n} w(v_i, v_{i+1}), && (4)
\end{aligned}$$

where $v_{n+1} = v_1$. The last sum is exactly the weight of the TSP tour induced by π.

1.2.5 Correctness of the Reduction

Since the terms $n \cdot M$ and $W_{\max} \cdot \frac{(n-1)n}{2}$ are constants independent of the vertex ordering, we have:

$$\mathrm{Cost}_{\mathrm{FOO}}(\pi) = K + \mathrm{Cost}_{\mathrm{TSP}}(\pi)$$

where $K = n \cdot M - W_{\max} \cdot \frac{(n-1)n}{2}$ is a constant.

This establishes a cost-preserving correspondence: the map that sends a FOO order to its induced cycle is a many-to-one (rotation, and in the undirected case also reversal) map onto the set of TSP tours. Consequently, an order is optimal for FOO iff its induced tour is optimal for TSP.

1.2.6 Complexity Analysis

The reduction requires:

- $O(|E|)$ time to compute $W_{\max}$
- $O(1)$ time to define the cost function
- Total time complexity: $O(|E|) = O(n^2)$ for a complete graph

Therefore, the reduction is polynomial-time computable.

1.2.7 Conclusion

We have demonstrated that:

1. TSP can be reduced to FOO in polynomial time
2. The reduction preserves optimal solutions
3. TSP is known to be NP-hard

By the transitivity of polynomial-time reductions, the Finding Optimal Order problem is NP-hard. $\square$

References

[1] Takuya Akiba, Yoichi Iwata, and Yuichi Yoshida. Fast exact shortest-path distance queries on large networks by pruned landmark labeling. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, page 349–360, New York, NY, USA, 2013. Association for Computing Machinery.