Mastering Routes: Traversing the History of the Travelling Salesman Problem

Neil Mehta

Abstract

The Travelling Salesman Problem (TSP) is a popular problem studied in the fields of combinatorial optimization and computer science. In a nutshell, the TSP asks: Given a list of locations, and the distances between each pair of locations, what is the shortest possible route that visits each location exactly once and returns to the location of origin? Emerging from the practical challenges faced by postal messengers in the 1930's, the problem continues to find real-world applications in the ever-expanding industry of e-commerce logistics. Notably, at the large-scale, companies like Amazon utilize TSP-inspired algorithms to optimize delivery routes and costs for their drivers, and conversely, individual tourists apply TSP principles to devise near-optimal tours for their foreign visits. This paper aims to highlight some of the mathematics involved in TSP optimization, including a proof of the problem's NP-hardness, along with an examination of the algorithms and mathematical constructs developed across the history of the problem, leading up to modern-day implementations and consequences.

1 Introduction

The Traveling Salesman Problem (TSP) stands as an archetypal challenge in the realm of combinatorial optimization, compared quite aptly to the story of Homer's Odyssey by Martin Groetschel. The problem is renowned for its complexity and practical significance. At its core, TSP tasks mathematicians and computer scientists with finding the shortest possible route that visits a set of given locations exactly once before returning to the starting point. Despite its seemingly straightforward premise, the TSP is categorized as an NP-hard problem, meaning that as the number of locations increases, the time required to find an optimal solution grows exponentially. A more mathematical intuition follows:

- **P**: P is a complexity class that represents the set of all decision problems (ie., problems whose answer is either "yes" or "no") that can be solved in polynomial time. That is, given an instance of the problem, the number of steps required to decide the answer can be modelled by some polynomial n^k, $k \in \mathbb{Z}^{\geq 0}$
- **NP**: NP is a complexity class that represents the set of all decision problems for which the instances where the answer is "yes" have proofs that can be checked in polynomial time.
- **NP-Complete**: NP-Complete is a complexity class which represents the set of all problems $X \in$ NP for which it is possible to "reduce" any other NP problem Y to X in polynomial time. More simply, we can solve Y quickly if we know how to solve X quickly.
- **NP-Hard**: These are the problems that are at least as hard to solve as the NP-complete problems. Note that NP-hard problems do not have to be in NP, and they do not have to have yes/no answers. The precise definition here is that a problem X is NP-hard, if there is an NP-complete problem Y, such that Y is reducible to X in polynomial time. Any NP-Complete problem is also NP-hard, so we can simplify this definition to the same as above: we can solve Y quickly if we know how to solve X quickly.

Theorem 1.1. *The TSP is NP-hard*

As explained above, the method to prove a problem is NP-hard is to show that solving the problem Y is the same as solving problem X, where X is an NP-hard or NP-Complete problem itself.

Proof

We begin by formulating a more mathematical statement of the TSP:

Given a complete weighted graph G with positive integer weights, find a Hamilton Cycle in G (ie., a route that visits each vertex exactly once, returning to the original vertex at the end) with minimum weight, where a weighted graph is a graph in which a number (the weight) is assigned to each edge.

To show that TSP is NP-hard, we must reduce another known and documented NP-hard problem to the form of the TSP (in other words, show that solving this other problem is the same as solving the TSP). The problem we will reduce is the Hamiltonian Cycle Problem, which asks if a graph G contains a Hamiltonian cycle:

Definition 1.2. Hamiltonian Cycle Problem: Let $G = (V, E)$ be an undirected graph. The Hamiltonian Cycle problem determines whether there exists a permutation p of the vertices $v_1, v_2, \ldots, v_n$ such that $(v_i, v_{i+1}) \in E$ for $i = 1, 2, \ldots, n-1$, and $(v_n, v_1) \in E$

The relation between the Hamiltonian Cycle Problem and the TSP is clear - both problems ask for a Hamiltonian cycle present within a graph. The difference between the two is that the Hamiltonian Cycle problem is a decision problem - its answer is yes/no, while the TSP asks for the **shortest** cycle within the complete graph.

As such, we can convert every instance of the Hamiltonian Cycle Problem, which consists of a graph $G = (V, E)$, into a Travelling Salesman problem, consisting of graph $G' = (V', E')$ and the maximum cost of G', called K. We will construct the graph G' in the following way:

For each edge $e \in E$, add the cost of edge cost$(e) = 1$. Connect the remaining edges, $e' \in E'$, that are not in the original graph G, each with a cost cost$(e') = 2$. Additionally, set $K = N$. That is to say, G' has edges in G, which has weight of 1, and edges not in G (but in the complete graph that is formed by the vertices in G).

The new graph G' can be constructed in polynomial time by just converting G to a complete graph G' and adding corresponding costs. This reduction can be proved by the following two claims:

1. Let us assume that the graph G contains a Hamiltonian Cycle (Figure 1), traversing all the vertices V of the graph. Now, these vertices form a TSP with cost $= N$ since it uses all the edges of the original graph having cost$(e) = 1$ and there are N edges in the cycle. And, since it is a cycle, it therefore returns back to the original vertex.

2. We assume that the graph G' (Figure 2) contains a TSP with cost $K = N$. The TSP traverses all the vertices of the graph returning to the original vertex. Now since none of the vertices are excluded from the graph and the cost sums to N, it must necessarily use all the edges of the graph present in E, with cost 1, hence forming a Hamiltonian cycle with the graph G.

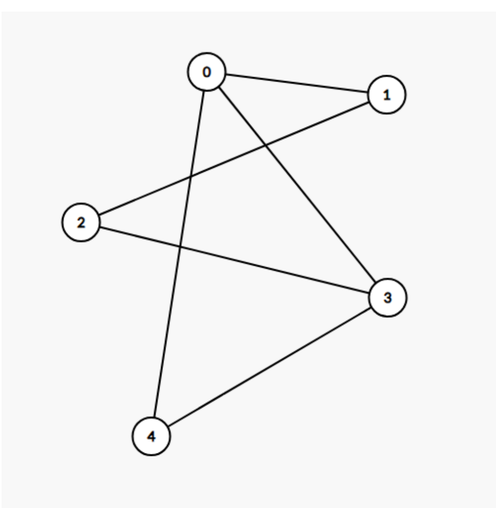

Figure 1: Graph of G for the Hamiltonian Cycle problem

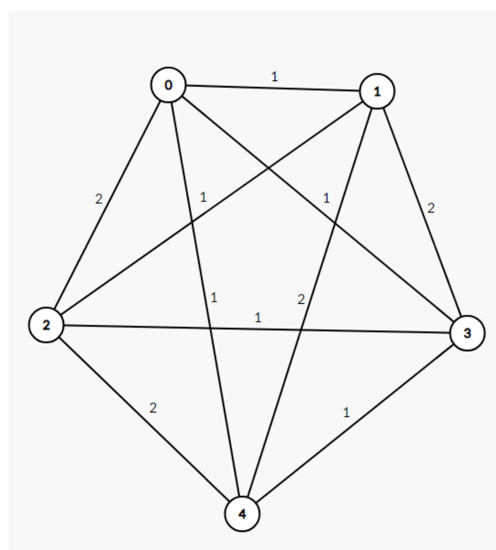

Figure 2: Graph of G' for the TSP

Since G' is constructed using the Hamiltonian cycle from G with edges of cost 1 and adding the remaining edges to complete the graph (with these edges costing 2), we know that G' certainly contains at least one Hamiltonian Cycle, in particular the cycle from G. Thus we can say that if G contains a Hamiltonian Cycle, then G' contains a TSP.
Hence, any instance of the TSP can be reduced to an instance of the Hamiltonian Cycle problem (in other words, we can solve any instance of the TSP if we can solve a similarly-constructed instance of the Hamiltonian Cycle Problem). Therefore, since the TSP reduces to an NP-Hard problem, the TSP itself is NP-Hard. $\square$

The history of the Traveling Salesman Problem traces back to the early 19th century, when mathematicians began grappling with questions of optimal route planning. However, it wasn't until the 1930s that Karl Menger, Hassler Whitney, and Merril Meeks Flood formally introduced the problem in the context of route optimization. Their work laid the groundwork for subsequent research and established TSP's relevance in various fields, including transportation, logistics, and computer science. In particular, the computer science field approach to the TSP involves leveraging linear programming, a powerful optimization methodology central to tackling TSP instances. Linear programming techniques enable the formulation of mathematical models that represent the constraints and objectives inherent in TSP instances, allowing for the systematic exploration of feasible solutions. Linear programming allows for the usage of powerful algorithms such as the simplex method and cutting planes in order to reduce time complexity of cycle optimization, both of which are discussed later in this paper.

Today, the Traveling Salesman Problem continues to captivate researchers and practitioners alike, driving ongoing exploration into its theoretical foundations and practical applications, while still employing and building upon the foundations laid by mathematicians of the mid-to-late 20^{th} century.

2 History

Imagine yourself as a delivery driver for Amazon, one of the world's foremost companies in international shipping. Each day, you take a Prime delivery van full of packages through 150 or more customer stops, following a predetermined route known to be the shortest on your GPS, starting and ending at a delivery depot. One day you begin to question the GPS - how can you be sure it truly is the shortest all-round route? How was the route even designed? Is there any path which can save even more time on these deliveries? These contemplations served as the fundamental motivation behind the creation of the TSP, nearly 100 years ago.

In 1930, Karl Menger was a mathematician studying algebra and dimensionality - more specifically, he was examining the length of a simple curve C in a metric space S. His findings led him to generalize his ideas into the following statement:

> For any finite subset X of a metric space, $\lambda(X)$ is the shortest length of a path through X

In other words, $\lambda(X)$ is a function representing a Hamiltonian path (ie., a path visiting each vertex of a graph exactly once) through a space S. In this case, the elements of S would be the vertices and $\lambda(X)$ would be the path connecting each element exactly once.

As his exploration into paths and graphs continued, Menger eventually posed the problem as what is recognized as the first rendition of the TSP to students in his Vienna workshop *Mathematisches Kolloquium*. The question was posed as follows:

> We denote by messenger problem (since in practice this question should be solved by each postman, anyway also by many travellers) the task to find, for finitely many points whose pairwise distances are known, the shortest route connecting the points. Of course, this problem is solvable by finitely many trials. Rules which would push the number of trials below the number of permutations of the given points, are not known. The rule that one first should go from the starting point to the closest point, then to the point closest to this, etc., in general does not yield the shortest route.[4]

Effectively, Menger asked for the *shortest Hamiltonian cycle* through the given points, while aware of the fact that the "**nearest-neighbour**" method was not necessarily the optimal solution.

Menger then spent the next few months as a guest lecturer at Harvard University, where he presented his results on arc lengths and shortest paths. In one of these lectures, Menger met Hassler Whitney, a Ph.D. researcher studying graph theory at the time. Whitney provided Menger with suggestions regarding his research on paths, and, in the following year, presented his own version of the TSP during a seminar talk at Princeton University.

Whitney posed the problem as finding the shortest route along the (at the time) 48 States of America. In the audience of one of his seminars was another Ph.D. researcher - Merril Meeks Flood, who then popularized the problem at the RAND Corporation in 1948, over 10 years after hearing it at Whitney's talk. As is the case for many other combinatorial optimization problems, the RAND Corporation played an important role in the research on the TSP. In 1949, Julia Robinson published a RAND Report containing what is believed to be the first mathematical reference using the term "traveling salesman problem":

> The purpose of this note is to give a method for solving a problem related to the traveling salesman problem. One formulation is to find the shortest route for a salesman starting from Washington, visiting all the state capitals and then returning to Washington. More generally, to find the shortest closed curve containing n given points in the plane.[4]

In 1954, the largest breakthrough in TSP research occurred. RAND researchers George Dantzig, Ray Fulkerson, and Selmer Johnson published a seminal paper titled "Solution of a large-scale traveling salesman problem" which introduced the method of **cutting planes** - something now considered a basic technique across all of combinatorial optimization, and whose math will be explained further in this paper. In essence, cutting planes provided a lower bound for optimal tour length when minimized over certain constraints. The paper did not provide an algorithm to solve the TSP, but does give a tour and proves that it is optimal with the help of constraints mentioned earlier. Cutting planes apply the **simplex algorithm**:

Definition 2.1. The simplex algorithm: an iterative procedure used to solve linear programming problems beginning with an initial feasible solution, and systematically traversing edges of the feasible region until the optimal solution (ie. maximum or minimum) is achieved. The algorithm works as follows:

1. **Initial Setup**:
    - Convert the linear programming problem into standard form.
    - Introduce slack variables for each inequality constraint to convert them into equality constraints.
    - Formulate an initial table representing the problem
2. **Select Entering Variable**
    - Choose the most negative coefficient in the objective function as the entering variable.
3. **Select Leaving Variable**
    - Compute the minimum ratios of the constants in the rightmost column to the corresponding coefficients in the column of the entering variable. The row with the minimum ratio corresponds to the leaving variable.
4. **Pivot Operation**:
    - Perform a pivot operation to make the entering variable basic and the leaving variable non-basic. Update the table accordingly.
5. **Repeat 2-4**
    - If the objective function is still not optimal (ie., there are still negative coefficients in the bottom row), repeat steps 2-4. Otherwise, the solution is optimal

6. **Read Solution**
   - The values of the basic variables in the final tableau correspond to the optimal solution.
   - The value in the bottom-right corner of the tableau is the optimal value of the objective function.

Definition 2.2. Cutting plane method: any of a variety of optimization methods that iteratively refine a feasible set or objective function by means of linear inequalities, termed *cuts* (ie., the inequalities *cut away* at the bounds of the solution, increasing optimality)

Essentially, cutting-plane inequalities trim away at the edges of the feasible solution region of the simplex algorithm, thereby providing a lower bound to the TSP. Below is a very simple example utilizing the cutting-planes method. Note that cutting planes solve a relaxed version of Linear Programming problems, converging to an integer solution. Hence, if the solution is already an integer, it is also a solution to the original problem:

Example 2.3.

$$\begin{aligned}&\text{Maximize } 2x_1 + 3x_2 \text{ such that:}\\ &\quad x_1 + 2x_2 \leq 4\\ &\quad 2x_1 + x_2 \leq 5\\ &\quad x_1, x_2 \geq 0,\ x_1, x_2 \in \mathbb{Z}\end{aligned}$$

Linear Programming Solution (Iteration 1):
Solving this linear relaxation, we find that the optimal solution is $x_1 = 1.25$ and $x_2 = 1.75$, with an objective value of $2 \cdot 1.25 + 3 \cdot 1.75 = 8.75$.

Introducing Cutting Planes:
We introduce cutting planes to strengthen the linear relaxation and obtain better bounds. We identify that the optimal solution must have integer values for x_1 and x_2. Let's consider the fractional part of the solution: $x_1 = 1.25$ and $x_2 = 1.75$.

First Cutting Plane:
Since the solution is fractional, we can introduce a cutting plane to restrict the solution space. One possible cutting plane is:

$$x_1 \leq 1$$

This additional constraint will cut off the fractional part of x_1, forcing it to take integer values less than or equal to 1.

Revised Linear Problem:
Now, our linear problem with the cutting plane becomes:

$$\begin{aligned}&\text{Maximize } 2x_1 + 3x_2 \text{ such that:}\\ &\quad x_1 + 2x_2 \leq 4\\ &\quad 2x_1 + x_2 \leq 5\\ &\quad x_1 \leq 1\\ &\quad x_1, x_2 \geq 0\end{aligned}$$

Linear Programming Solution (Iteration 2):
Solving this revised linear problem, we find that the optimal solution is $x_1 = 1$ and $x_2 = 1.5$, with an objective value of $2 \cdot 1 + 3 \cdot 1.5 = 6.5$.

Second Cutting Plane:
Since the solution is still fractional for x_2, we introduce another cutting plane:

$$x_2 \leq 1$$

This additional constraint will cut off the fractional part of x_2, forcing it to take integer values less than or equal to 1.

Final Linear Problem with Cutting Planes:

$$\begin{aligned}
&\text{Maximize } 2x_1 + 3x_2 \text{ such that:}\\
&\quad x_1 + 2x_2 \leq 4\\
&\quad 2x_1 + x_2 \leq 5\\
&\quad x_1 \leq 1,\ x_2 \leq 1\\
&\quad x_1, x_2 \geq 0
\end{aligned}$$

Final Solution (Iteration 3):
Solving this revised linear relaxation with both cutting planes, we find that the optimal solution is $x_1 = 1$ and $x_2 = 1$, with an objective value of $2 \cdot 1 + 3 \cdot 1 = 5$. $\square$

Using this method, Dantzig, Fulkerson, and Johnson were able to find the shortest tour (ie., Hamiltonian cycle) along cities chosen in the 48 U.S. states and Washington, D.C. This research forms the basis for almost all further enhancements on large-scale traveling salesman problems.

3 Modern Day Problems and Adaptations

More recently, improvements to TSP-optimizing algorithms have greatly evolved, allowing for problems of greater magnitude and complexity to be solved as well. One such example is an improvement to the simplex algorithm, known as the **branch and bound** method.

Definition 3.1. Branch and Bound
Branch and bound is an integer programming technique - it only works on integer values. The method works by branching integer variables into two sets, upper and lower bounds of that variable, and then checking the obtained solution against the feasibility (ie., correctness) objective function. One can imagine the branch and bound algorithm as a binary search tree algorithm, if familiar with the computer science concept.

However, research on the Traveling Salesman problem has not only advanced its own solutions and optimization techniques, but has also catalyzed the emergence of spin-off problems and branches of mathematics. Indeed, TSP has spurred research even outside of mathematics, providing a baseline for genetic algorithm research - a method for solving both constrained and unconstrained optimization problems based on the biological phenomenon natural selection.

Moreover, exploration into TSP have led to the realization of related combinatorial optimization problems like the Vehicle Routing Problem (VRP), which seeks to optimize the delivery routes for multiple vehicles - similar to the above-mentioned Amazon delivery driver perspective. Companies such as Amazon, FedEx, UPS, etc. have invested uncountable resources into improving their delivery algorithms because, in a world with ever-expanding demand for "shipping and handling", even a 2 percent optimization over the current delivery times would save hundreds of thousands of dollars, hours, and even tonnes of greenhouse gas emissions.

One company that uses a modern variation of the TSP is Wal-Mart. During the company's rise through the ranks of retail stores, it's founder, Sam Walton, employed a "hub and spoke" method in order to improve the company's supply chain logistics. Wal-Mart establishes several distribution centers and stores within a limited radius of a single supply warehouse - and in doing so, the company significantly cuts down shipping and transportation costs from 5 percent for normal customers to 3 percent. As explained above, this 2 percent improvement resulted in a catapulting of Wal-Mart to the very top of the retail industry, where they hold this position to this very day. The "hub and spoke" method Wal-Mart employs also goes by a different name: the **Clustered Travelling Salesman Problem (CTSP)**.

Definition 3.2. Clustered Travelling Salesman Problem
The CTSP is an extension of the TSP where the set of cities is partitioned into clusters, and the salesman has to visit the cities of each cluster consecutively. It is well known that any instance of CTSP can be transformed into a standard instance of the TSP, thus making it also NP-Hard.

More formally, the CTSP is defined on a complete graph $G = (V, E)$, where $V = \{v_1 \dots v_n\}$ is the vertex set and $E = \{(v_i, v_j) : v_i, v_j \in V, i \neq j\}$ is the edge set. Each edge is assigned a cost c, and V is partitioned into m disjoint clusters such that

$$V = V_1 \cup V_2 \cup \ldots \cup V_m, \text{ with } V_i \cap V_j = \emptyset \quad \forall i \neq j$$

with the aim of finding a Hamiltonian cycle in G such that the cost is minimized.

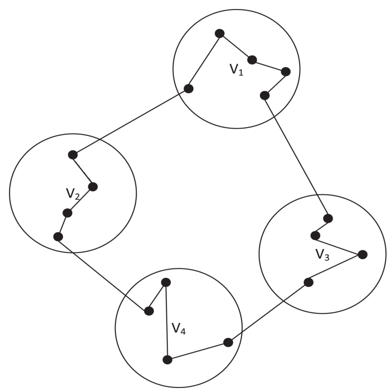

Figure 3: Example diagram of the CTSP

4 Conclusion

In short, the Travelling Salesman Problem plays a prominent role in combinatorial optimization and computer science, stemming from the humble beginnings of postal deliveries in the 1930s. The construction of progressively powerful algorithms through the years has developed a cornerstone of research and practical application, finding relevance in fields as diverse as e-commerce logistics, transportation, biological evolution and many others. Several important techniques for solving combinatorial optimisation problems were developed using the TSP as an example application, including (but not limited to) cutting planes in linear programming and the simplex algorithm, which in turn have led to more modern and efficient algorithms such as the branch and bound method.

The impact of TSP research extends beyond its own domain, catalyzing the emergence of spin-off problems like the Vehicle Routing Problem. Companies like Amazon and Wal-Mart have leveraged TSP-inspired algorithms to optimize their delivery routes, resulting in substantial cost savings and efficiency gains. As researchers and practitioners continue to push the boundaries of optimization theory and application, the legacy of the TSP serves as a beacon of inspiration, guiding future innovations and discoveries in the quest for optimal solutions in a complex and interconnected world.

5 Bibliography

1. Figueroa, J., et al. (2012). A Modern Application of the Traveling Salesman Problem: Understanding Walmart. web.williams.edu/Mathematics/sjmiller/public_html/377Fa16/projects/MATH416_Walmart_FigueroaMillerOkimotoFinal_121212.pdf.

2. Grötschel, Martin, and Manfred Padberg. (2001). "The Optimized Odyssey." Zuse Intstitute Berlin, AIROnews, www.zib.de/groetschel/pubnew/paper/groetschelpadberg2001a.pdf.

3. Lu, Y., et al. (2022). Solving the clustered traveling salesman problem via traveling salesman problem methods. In PeerJ Computer Science (Vol. 8, p. e972). PeerJ. https://doi.org/10.7717/peerj-cs.972

4. Schrijver, A. (2005). On the History of Combinatorial Optimization (Till 1960). In Discrete Optimization (pp. 1–68). Elsevier. https://doi.org/10.1016/s0927-0507(05)12001-5