**BYD 还不睡觉**

**Abstract:**

The main purpose of this paper is to compare already existing methods of TSP…..

Introduce the idea of different method

**Introduction：**

The origins of the Traveling Salesman Problem date back to the 18th century, with initial discussions conducted by mathematicians Hamilton and Kirkman. However, it was Karl Menger in Vienna and Harvard who advanced a more generalized version of the TSP[1] . Considering a collection of cities with their cost of traveling (or distance ) between each pair, the Traveling Salesman Problem (TSP) seeks to determine the most efficient route for visiting all cities and returning to the starting location while minimizing the total travel cost [1]. A lot of good algorithms have been developed for the study of the TSP. We can characterize them into two categories: exact algorithm and heuristic algorithm.

Exact algorithm: No efficient algorithms are known for this class. It is assumed that any algorithm designed to solve these problems exactly requires exponentially many steps[2]. These algorithms are very slow when applied to the large scale of the problem. However, they will return an optimal solution, and compared to the heuristic algorithm, little mathematical proof exists for the assumption.

Heuristic algorithm: These algorithms trade optimality for speed and simplicity. We are interested in the idea of these algorithms.

Even though both exact and heuristic methods have demonstrated success, there is no single technique that is guaranteed to outperform others in every situation. Different methods depend on the setting of the problem (size) . This absence of a universally ideal mathematical approach encourages the comparative analysis of various algorithms: by testing their performance on different data sets. This will provide a clearer understanding of the trade-offs and potential advantages of different methods, allowing us to understand the advantages and disadvantages of using the different approach.

**Mathematical model for the Problem:**

We begin at one of the n cities, proceed to visit each of the remaining cities sequentially, and ultimately return to our starting location. The core optimization challenge is to determine a route with the shortest total distance. To achieve this, the distances between each pair of cities are provided in a table or matrix format. [5]

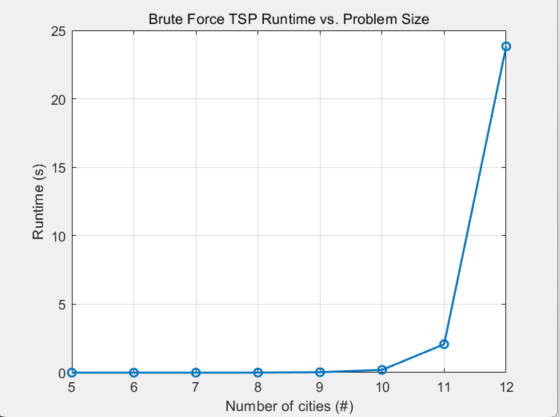
d(i,j) with $0\leq i,j \leq m$

Hamiltonian cycle

**ALGORITHMS:**

**Bruteforce force method :**

The brute-force approach is the most basic algorithm to solve the Traveling Salesman Problem (TSP) precisely. It examines every possible tour individually, calculates the total distance of each tour, and determines the optimal tour by identifying the minimum among these distance values through comparisons. Therefore, this algorithm will always give the optimal solution.

For example if we have 5 cities A, B, C, D, E and their location with knowing coordinates. Then we list out all the possible routes, There are 5! Many combinations of routes. In our settings, if we fix A as the initial city we start from, then We have 4! different ways to visit n cities by a tour. Next, we analyze the total distance for every route to identify the shortest one. Yet, as the number of cities increases, the possible routes grow uncontrollably . We need to calculate (n − 1)! permutations of the route. Even with just sixteen cities, it could take 20 years to find the optimal route[2]. 

**Dynamic programming：**

For 16 cities, The algorithm reduces 20 years to 4.6 hours to find the optimal route[2].

**Integer linear programming：**

**Nearest neighbor :**

The nearest neighbor algorithm is a quick and straightforward approach for creating a TSP tour. The concept is to start from any chosen city, then continually travel to the unvisited city with the lowest travel expense until every city has been visited, cand finally return to the starting point[3]. Because the concept is straightforward, we might achieve a quick algorithm, but the precision of the optimization may decrease. it is easy to implement and quick to run with a worst case time complexity is O(n^2)[4].

**Comparative study:**

**Current trend for TSP:**

Bibliography:

[1] Davendra, D. (2010). *Traveling salesman problem, theory and applications*. INTECH.

https://books.google.com/books?hl=en&lr=&id=gKWdDwAAQBAJ&oi=fnd&pg=PR11&dq=Traveling+salesman+problem,+theory+and+applications.+INTECH.&ots=aacyW88cK5&sig=nqbEJHHqg-q9UUQIbzZJ23Rsh6I#v=onepage&q=Traveling%20salesman%20problem%2C%20theory%20and%20applications.%20INTECH.&f=false

[2 ]Näher, S. (2019b). *Algorithms unplugged*. Scholars Portal.

<https://link.springer.com/chapter/10.1007/978-3-642-15328-0_40#citeas>

[3]Hurkens, C. A., & Woeginger, G. J. (2003). On the nearest neighbor rule for the traveling salesman problem. *Operations Research Letters*, *32*(1), 1–4. <https://doi.org/10.1016/s0167-6377(03)00093-2>

[4]Neoh, A., Chen, H., & Chase, C. (2020). An Evaluation of the Traveling Salesman Problem.

<https://scholarworks.calstate.edu/downloads/xg94hr81q>

[5] Author links open overlay panelPatchara Kitjacharoenchai a, et al. “Multiple Traveling Salesman Problem with Drones: Mathematical Model and Heuristic Approach.” *Computers & Industrial Engineering*, Pergamon, 10 Jan. 2019,

[www.sciencedirect.com/science/article/abs/pii/S0360835219300245](http://www.sciencedirect.com/science/article/abs/pii/S0360835219300245)