Week 8 Discussion

Available until May 10, 2023 2:59 AM. **Submission restricted after availability ends.**

Must post first.

The traveling salesperson problem is a harder problem than Dijkstra's single-source shortest path problem. In other words, the typical Greedy algorithm approach does not work for this problem. It is even harder than the all-points shortest path algorithm implemented with Floyd's algorithm. Give an example of a graph that shows that the path that would be chosen by relying on shortest-path information by choosing the closest vertex each time isn't sufficient to find the shortest circuit. What makes this problem harder? Why are the straight forward approaches to this problem exponential?

Introduction:

The Traveling Salesperson Problem (TSP) is a well-known NP-hard optimization problem that seeks to find the shortest possible path that visits all nodes of a graph exactly once and returns to the starting node. In this answer, we will discuss the difficulties involved in solving the TSP and explore different methods for solving it.

The Challenge:

The challenge in solving the TSP is that it is a combinatorial optimization problem with a large search space. The number of possible solutions grows very quickly with the number of nodes in the graph, making it difficult to solve for larger graphs.

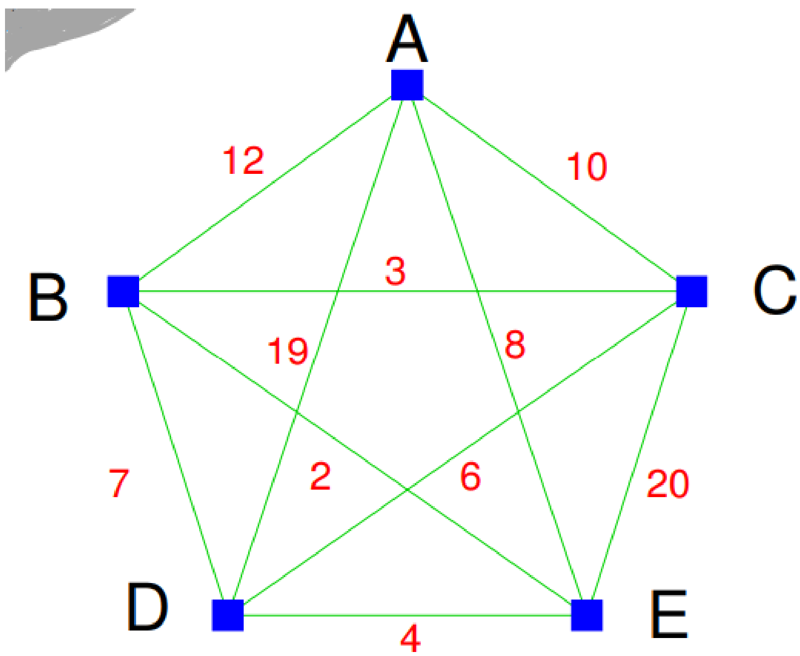
Difference between Hamiltonian Cycle and TSP:

While the Hamiltonian Cycle is a similar problem, it only requires finding one path that visits all nodes exactly once, whereas the TSP requires finding the shortest path that visits all nodes exactly once. The TSP is more difficult because there may be many paths that visit all nodes, but only one is the shortest.

Why Greedy Algorithm Fails:

A common approach for solving optimization problems is the Greedy Algorithm, where at each step, the algorithm chooses the locally optimal solution. However, this approach is not effective for the TSP since choosing the closest node at each step does not guarantee that we will find the shortest path that visits all nodes exactly once.

Example: Consider the following TSP graph:



In here the choosing the closest vertex each time is not sufficient enough to find the shortest Hamiltonian path because we have multiple of these paths so when we choose the closest next vertex in each path, we will have multiple path the closest in each next vertex does not assure that it will lead to the overall optimized paths. We need a specific algorithm known as Dynamic Programming where we need to optimize each substructure in each step.

Approach:

Dynamic Programming A more effective approach for solving the TSP is Dynamic Programming, where we break the problem down into smaller subproblems and solve them recursively. We start by defining a function f(S, i) that gives the length of the shortest path that visits all nodes in set S exactly once and ends at node i. We can then use this function to solve the larger problem by considering all possible sets of nodes and all possible ending nodes.

Time Complexity:

While Dynamic Programming can be an effective approach for solving the TSP, the time complexity of the algorithm can still be exponential in the worst case. A brute-force approach that checks all possible permutations of nodes would have a time complexity of (N-1)! where N is the number of nodes, making it impractical for larger graphs.

Conclusion:

In conclusion, the Traveling Salesperson Problem is a challenging optimization problem that requires finding the shortest path that visits all nodes of a graph exactly once. While Greedy algorithms fail in this case, Dynamic Programming can be an effective approach for solving the problem. However, the time complexity of the algorithm can still be exponential in the worst case, and brute force approaches are not practical for larger graphs.

References:

1. M. Held, R. M. Karp. "The traveling-salesman problem and minimum spanning trees", Operations Research, 18(6):1138-1162, 1970.
2. C. H. Papadimitriou, K. Steiglitz. "Combinatorial Optimization: Algorithms and Complexity", Prentice-Hall, 1982.
3. T. Cormen, C. Leiserson, R. Rivest, C. Stein. "Introduction to Algorithms", MIT Press and McGraw-Hill, 2009.
4. J. E. Hopcroft, R. Motwani, J. D. Ullman. "Introduction to Automata Theory, Languages, and Computation", Addison-Wesley, 2001.
5. N. Christofides. "Worst-case analysis of a new heuristic for the travelling salesman problem", Report 388, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA, 1976.