Held-Karp Algorithm to Approximate Solutions to the Traveling Salesperson Problem

Drake Foltz, Kailean O’Keefe, Tsung Huang, Carson Hinckley

Received 15 April 2019

CS 312-001

Professor Ryan Farrell

BYU, Computer Science

Abstract

The Traveling Salesperson Problem has long been one of the most visible NP-Complete problems in the fields of computer science and mathematics. What follows is an examination and implementation of a Held-Karp algorithm for quickly approximating solutions to this problem. This dynamic programming algorithm will be compared to a greedy approximation, random path, and a branch-and-bound algorithm. Our intention is to show that the dynamic Held-Karp algorithm’s relative optimization in terms of accuracy and time/space complexity.

1. Introduction

The Traveling Salesperson Problem (TSP) is especially confounding to scientists, because of the cognitive gap between understanding the problem and being able to efficiently find the true solution. Included in this report will be a explanation of the greedy algorithm for approximating a solution to the traveling salesperson problem, and a further exploration into the Held-Karp dynamic programming algorithm which attempts a more accurate approximation. We will be comparing the time and space complexity of these implementations of these two algorithms, as well as providing empirical evidence supporting

We will first discuss our implementation of a greedy algorithm which finds a temporary solution to the TSP problem. We use this solution, as well as the solution provided by a random tour, in establishing a baseline of comparison. After we discuss our implementation of the Held-Karp, we will present a comparison of results between these three algorithms as well as our previous implementation of the Branch and Bound algorithm. After this presentation we will give a short conclusion, followed by the resources that we have used.

2.1 Greedy algorithm

In order to properly run our greedy algorithm, we first obtain the result from the provided random tour algorithm. We use this result as an upper bound on updating our current, lowest cost result. Then, for each city in the graph and in the time allotted, we build a path, always utilizing the edge which requires the least cost. The only other requirement for building the path is that it makes a Hamiltonian circuit, or that no cities are repeated in the path except the city used as the start and end position. Once the path is completely calculated, we then update our best solution so far if the new path has a lower cost than the best solution. After all this as run for all cities and within the time allotted, then we return our best solution so far. This function uses two nested loops, each bounded by the number of cities, and then a function call inside the inner loop which also runs through all cities. These all result in a complexity of O(n³), with n being the number of cities.

1.1.1 Subsubsection heading. In in nunc. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Donec ullamcorper fringilla eros. Fusce in sapien eu purus dapibus commodo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Cras faucibus condimentum odio. Sed ac ligula. Aliquam at eros.

Etiam at ligula et tellus ullamcorper ultrices. In fermentum, lorem non cursus porttitor, diam urna accumsan lacus, sed interdum wisi nibh nec nisl. Ut tincidunt volutpat urna. Mauris eleifend nulla eget mauris. Sed cursus quam id felis. Curabitur posuere quam vel nibh. Cras dapibus dapibus nisl. Vestibulum quis dolor a felis congue vehicula. Maecenas pede purus, tristique ac, tempus eget, egestas quis, mauris. Curabitur non eros. Nullam hendrerit bibendum justo. Fusce iaculis, est quis lacinia pretium, pede metus molestie lacus, at gravida wisi ante at libero. Quisque ornare placerat risus. Ut molestie magna at mi. Integer aliquet mauris et nibh. Ut mattis ligula posuere velit.

Nunc sagittis. Curabitur varius fringilla nisl. Duis pretium mi euismod erat. Maecenas id augue. Nam vulputate. Duis a quam non neque lobortis malesuada. Praesent euismod. Donec nulla augue, venenatis scelerisque, dapibus a, consequat at, leo.

Pellentesque libero lectus, tristique ac, consectetuer sit amet, imperdiet ut, justo. Sed aliquam odio vitae tortor. Proin hendrerit tempus arcu. In hac habitasse platea dictumst. Suspendisse potenti. Vivamus vitae massa adipiscing est lacinia sodales. Donec metus massa, mollis vel, tempus placerat, vestibulum condimentum, ligula. Nunc lacus metus, posuere eget, lacinia eu, varius quis, libero.

Aliquam nonummy adipiscing augue. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas.

Acknowledgements

Proin pharetra nonummy pede. Mauris et orci. Aenean nec lorem. In porttitor. Donec laoreet nonummy augue. Suspendisse dui purus, scelerisque at, vulputate vitae, pretium mattis, nunc. Mauris eget neque at sem venenatis eleifend. Ut nonummy.

References

1. Surname A, Surname B and Surname C 2015 *Journal Name* **37** 074203
2. Surname A and Surname B 2009 *Journal Name* **23** 544