Ryan Wade

Math 307

Kumar

28 April 2019

Efficiently Finding an Optimal Solution to the Traveling Salesman Problem

Perhaps the most famous NP-Complete problem in discrete mathematics and computer science is the Traveling Salesman Problem, a problem where a computer is given the coordinates for some number of cities and has to find the shortest route between them using as little computing power as possible. Initially, this doesn't sound like that difficult of a problem, until you realize the computational complexity of solving this problem by brute force is n!. So if you were to find the distance between just 20 cities, the computer would need to compute the distances between every combination of cities would be 19!, equating to a mind bending 121,645,100,000,000,000 intercity distances. There are other algorithms for finding a solution, however, the solutions generated will not be exact, though they use exponentially less compute power than brute force methods. This project will comparing the differences between the Branch and Bound brute force algorithm and the Christofides algorithm using a hypothetic situation in the world of Game of Thrones as an example.

The Branch and Bound algorithm will find the most optimal solution every time, but with a worst case complexity of O(n!). The pseudocode for Branch and Bound is below:

```
While (there remain active nodes) {
      Select an active node j and mark it as inactive Let x(j) and zLP(j)
      denote the optimal solution and objective of the LP relaxation of
Problem(j).
      Case 1: z* ≥ zLP(j)
            Prune node j
      Case 2: (z* < zLP(j)) and (x(j) is feasible to compute)
            Replace the incumbent by x(j)
            Prune node j
      Case 3: (z* < zLP(j)) and (x(j) is not feasible to compute)
            Mark the direct descendants of node j as active
}
```

In simpler terms, the algorithm keeps track of the current shortest route from the starting location and "prunes" nodes (marking them inactive) when it believes the path up to that node is the shortest it can be (Case 3 in the pseudocode). However sometimes that isn't actually the shortest path and it has to re-mark nodes as active (Case 2 in the pseudocode). Branch and Bound continues iterating until all nodes are pruned and thus, the problem is solved.

The Christofides algorithm is a 1.5-approximation algorithm, so it won't find the shortest route, but it has a complexity of only $O(n^3)$ compared to brute force which is O(n!). The pseudocode for Christofides algorithm is below:

1. Create a minimum spanning tree *T* of *G*.
2. Let *O* be the set of vertices with odd degree in *T*. By the handshaking lemma, *O* has an even number of vertices.
3. Find a minimum-weight perfect matching *M* in the induced subgraph given by the vertices from *O*.
4. Combine the edges of *M* and *T* to form a connected multigraph *H* in which each vertex has even degree.
5. Form an Euler circuit in *H*.
6. Make the circuit found in previous step into a Hamiltonian circuit by skipping repeated vertices

After finding a minimum spanning tree from the given coordinates, it finds the closest vertex of odd degree (so any vertex that is either connected to only one other vertex or more than 2) that doesn't already have a match and connects them, making all vertices of even degree. This set is combined with the other half of even degree vertices, creating a multigraph through which a

Euler circuit is then formed. Finally, a Hamiltonian circuit is created from the Euler circuit by removing any repeated vertices.

The two algorithms' properties will be shown through an example that takes place at the end of Game of Thrones Season 7. The White Walkers and their undead army have just overtaken The Wall separating Westeros from The Land of Always Winter and are quickly marching down into Westeros, but leaders Daenerys Targaryen, Jon Snow, and Sansa Stark worry they don't have enough Dragon Glass (one of the only ways they know how to kill a White Walker) stocked at Winterfell. After Cerci betrayed her brother (and former lover) Jaime Lannister, he decides to share some intel that just might help them win the war. Years ago, Jaime learned that Cerci hid many stores of wildfire throughout Westeros and Essos and drew their locations on a map which Jaime copied before leaving for Winterfell. It's known that fire is an effective weapon against the walkers and undead, so it is up to Daenerys to fly to each location on her dragon in the shortest time possible.

The following maps were generated by a Java program written to plot randomly generated coordinates on top of a map of Game of Thrones and plot the shortest route computed by both the Branch and Bound and Christofides algorithms. So that the program only generates coordinates on land, it takes in the RGB values of the pixel at the location of a prospective coordinate and accepts it is that pixel isn't dark blue (aka not in the sea). Figure 1 shows the Branch and Bound algorithm solution and Figure 2 shows the solution generated by Christofides algorithm; stores of wildfire are marked by green flames and Winterfell is marked by a castle.

Figure 1 -
Branch and Bound Solution - 20 points

Figure 2 -
Christofide Solution - 20 points

The above iteration of the Branch and Bound solution took 68.9 seconds to compute the shortest path of 24,991.2 miles while the Christofide solution only took *0.01 seconds* to compute a path of 25,699.284 miles. Below is a table showing the results of 5 different instances of the above problem, all of them using 20 points.

#	BnB *t*	Christofide *t*	BnB Distance	Christofide Distance	How many times faster Christofides is	How many times longer Christofides is
1	68.918 seconds	0.01 seconds	24,991 miles	25,699 miles	6,891x faster	1.028x longer
2	2.149 seconds	0.004 seconds	29,073 miles	31,072 miles	537x faster	1.068x longer
3	12.809 seconds	0.006 seconds	27,087 miles	41,027 miles	2,134x faster	1.514x longer
4	19.322 second	0.01 seconds	27,726 miles	32,946 miles	1,932x faster	1.188x longer
5	0.382 seconds	0.002 seconds	26,948 miles	31,377 miles	191x faster	1.16xx longer

The time it took for Branch and Bound to find the solution for 20 coordinates ranged from less than a second to over a minute. That inconsistency is explained by the fact that the algorithm basically has to guess the shortest path, so sometimes it gets lucky and finds the path quickly, other times it doesn't. For most of them, the route generated by Christofides algorithm was only 5%-20% longer than BnB, except for the 3rd instance in which the route was more than 50% longer than the shortest path.

So if the Christofide algorithm can generate those solutions in fractions of a second, how fast would it take to solve 1000 points? Well in the case shown by Figure 3, only 0.267 seconds.

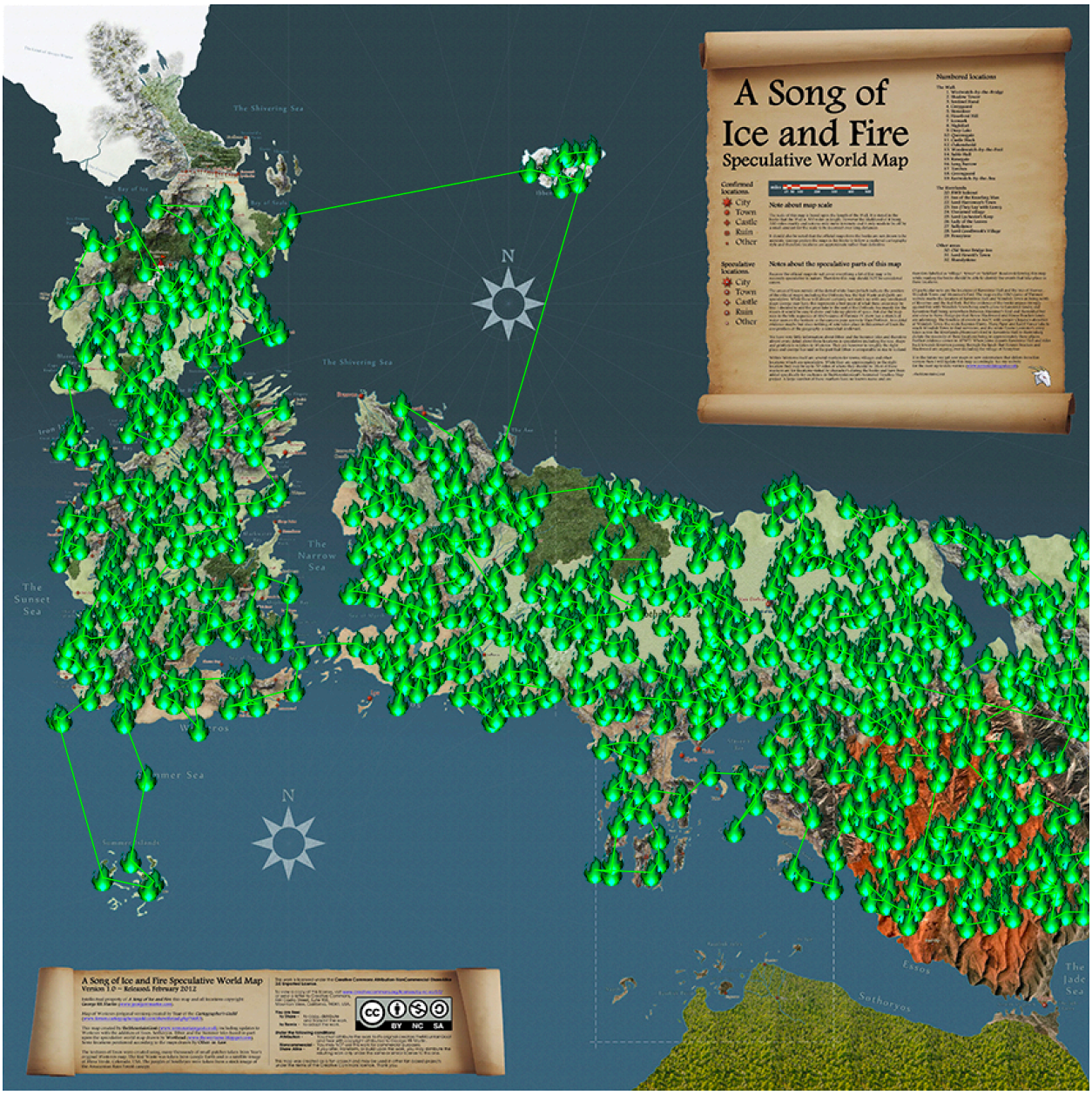

Figure 3 -
Christofide Solution - 1000 points

The aim to reduce the complexity required to solve the traveling salesman problem has many applications in which there are thousands or more points to consider. Applications range from finding the shortest mail route through a country to finding the shortest path when drilling circuit boards and even gathering precise information of crystalline structures through X-Ray crystallography (Matai). To put into perspective just how computationally demanding this problem is, the current record for the largest solved instance of the traveling salesman problem is 85,900, achieved by Bell Laboratories in 2006 when they successfully found the shortest distance for a laser to travel sculpting a computer chip (Operations Research Letters). One could only imagine all of the possible applications to surface as technology advances, but for now the only two choices are either to settle for approximate solutions, or to wait months or years for todays computers to solve larger problems.

Works Cited

"Certification of an Optimal TSP Tour through 85,900 Cities." *Operations Research Letters*, North-Holland, 15 Oct. 2008, www.sciencedirect.com/science/article/pii/S0167637708001132.

Matai, Rajesh, et al. *Traveling Salesman Problem: an Overview of Applications, Formulations, and Solution Approaches*. INTECH Open Access Publisher, 2010.

Haque, Albert. "TSP Hueristics - Christofides Algorithm Implementation." *Github*, 1 Dec. 2013, github.com/faisal22/Christofides.

McLane, Foster. "A Branch and Bound Solution to the Travelling Salesman Problem." *Github*, 31 Oct. 2013, github.com/fkmclane/TSP.