1 Christofide

1.1 Background

The Christofides algorithm improves on the 2-approximation algorithm, providing a worst case ratio of 3/2 at the cost of increasing the running time to $O(n^3)$ [1].

It can be summarized with the following steps:

- Using the entire graph, build a minimum-spanning tree Make a set of vertices
- Make a set of vertices with odd degree
- From the set of vertices with odd degree, make a minimum weight matching
- Add the minimum weight matching to the minimum-spanning tree
- Find a Euler tour from the graph made by combining the minimum spanning tree with minimum weight matching
- Remove repeated vertices, thus turning it into a Hamiltonian circuit [1]

Step one, getting the minimum spanning tree, makes it such that we are working with a new graph that has eliminated some of the higher cost edges without getting rid of any of the vertices we must visit (recall all must be visited). Next, we get the set of vertices with an odd degree which, by the handshaking lemma, must be an even number of vertices [2]. Finding a minimum weight matching on these and adding it to the minimum spanning tree ensures that the shortest of edges will be available when finding the tour. Finding a minimum weight matching also ensures each vertex has enough edges to avoid having to reuse edges in order to get off a vertex. In other words, it ensures a Euler tour can be found on this subgraph that includes all the lowest-weight edges. After finding a Euler tour, we can simply remove repeated vertices because each vertex can only be visited once, and in the actual map all vertices can be reached. Due to the triangle inequality, removing repeated vertices doesnt increase weight. [2] Tests have shown that the algorithm tends to place itself 10% above the Held-Karp lower bound[1].

The key difference between it and the 2-approximation algorithm is that, rather than simply duplicating edges to ensure an Euler cycle, it creates a minimum weight matching on the vertices with odd degree and combines it with the minimum spanning tree [1].

1.2 pseudocode

Below in the pseudocode for the algorithm. Note that it combines the step of finding a minimum weight matching with the one that combines it with the minimum spanning tree.

Algorithm 1 Christofide

```
1: procedure CHRISTOFIDE(\{G = (V, E),\ s\})
2:     for v ∈ V do
3:         v_key ← ∞
4:         v_parent ← ∅
5:     end for
6:     s_{key=0}
7:     while |k| do
8:     end while
9: end procedure
```

1.3 Works Cited