Christofides Algorithm

These lecture notes describe the polynomial-time algorithm developed by Nicos Christofides that approximates the solution to the metric travelling salesperson problem by providing a Hamiltonian cycle whose weight is at most $\frac{3}{2}$ of the optimal solution [2]. Section 1 provides all the relevant definitions and facts we treat as given because they have been proven in previous lectures. Section 2 outlines the algorithm. Section 3 proves the algorithm’s correctness and discusses its complexity.

1 Definitions and Given Facts

Definition 1.0.1 A Hamiltonian Cycle is a cycle that visits every vertex of a graph exactly once and then returns to the start vertex.

Definition 1.0.2 The Traveling Salesman Problem (TSP) is the task of finding a minimum weight Hamiltonian cycle in a complete weighted graph.

Definition 1.0.3 A complete graph satisfies the Triangle Inequality if for all vertices $i, j, k$, $w_{i,k} \leq w_{i,j} + w_{j,k}$.

Definition 1.0.4 The Metric Traveling Salesman Problem (MTSP) is a special case of TSP that takes as input a graph that satisfies the triangle inequality.

Definition 1.0.5 A Spanning Tree $S$ of a graph $G$ is a connected subgraph of $G$ with no cycles and containing all vertices of $G$. The Minimum Spanning Tree is the tree $S$ such that $w(S) \leq w(S')$ for all spanning trees $S'$, where the weight of a spanning tree is the sum of the weights on its edges.

1Thanks to Duc Nguyen for scribing a previous version of these notes
Definition 1.0.6 An Eulerian Tour is a cycle that uses every edge of a graph exactly once.

Fact 1.0.7 In a complete graph, the weight of a minimum spanning tree is less than the weight of the minimum-weight Hamiltonian cycle. See previous lecture notes for the proof.

Fact 1.0.8 In a complete graph satisfying triangle inequality, for any cycle $C$ that visits every vertex at least once, there exists a Hamiltonian cycle whose weight is at most that of $C$. Given $C$, such a Hamiltonian cycle can be found in linear time. See previous lecture notes for the proof.

2 The Algorithm

Input: A complete weighted graph $G$ that satisfies the triangle inequality. Figure 1 presents our running example for this algorithm.

Output: A Hamiltonian cycle on $G$ whose weight is at most $\frac{3}{2}$ of the optimal solution to the TSP problem. See Figure 2 for the possible output of the algorithm when applied to the graph on Figure 1.

Step 1: Find a minimum spanning tree $T$ of $G$. According to 1.0.7, $w(T) <$
Figure 2: Outlined in red is a possible output of the Christofides Algorithm on the graph shown in Figure 1

$OPT_G$, i.e. the weight of $T$ is less than the weight of the solution to MTSP on $G$. Figure 3 shows a minimum spanning tree for our running example.

Step 2: Let $G'$ be the subgraph of $G$ that only includes the vertices whose degree in $T$ is odd. In Section 3.1, we show that $G'$ must have an even number of vertices. Compute the minimum weight matching $M$ of $G'$. In Section 3.2, we show that $w(M) \leq \frac{OPT_G}{2}$. See Section 3.3 for a discussion of this step’s complexity. Figure 4 shows the minimum matching corresponding to the tree in Figure 3.

Step 3: Let $H$ be the graph that includes all edges in $T$ and $M$. Figure 5 shows what $H$ would look like for our running example. Note that all vertices in $H$ have even degree - those vertices that have odd degree in $T$ have exactly one more adjacent edge in $H$. In Section 3.4, we show that there exists an Eulerian Tour for any graph with no odd-degree vertices and that the tour can be computed in polynomial time. Compute an Eulerian Tour $E$ for $H$.

Step 4: Note that $w(E) \leq w(H) \leq w(T) + w(M) \leq \frac{3}{2}OPT_G$. According to 1.0.8, there must therefore be a Hamiltonian cycle in $H$, whose weight is guaranteed to be less than $\frac{3}{2}$ times the solution to the MTSP. Output this Hamiltonian cycle.
Figure 3: Highlighted edges form the minimum spanning tree for the graph in Figure 1

Figure 4: Highlighted edges form the minimum weight matching corresponding to the spanning tree in Figure 3
3 Correctness and Complexity

To prove the correctness of the Christofides algorithm and its polynomial runtime, we must prove the following four facts, as outlined above in the algorithm’s description.

3.1 Fact 1: Number of vertices with odd degree

Claim 3.1.1 In any graph, the number of vertices with odd degree is even.

Proof: For any undirected graph, the degree of a vertex is the number of edges incident to that vertex. Hence, every edge contributes exactly 2 to the sum of degrees of all vertices, which is, therefore, an even number. Let \(Odd\) be the set of odd-degree vertices and \(Even\) be the set of even degree vertices. We have that \(\sum_{v \in Odd \ deg(v)} + \sum_{v \in Even \ deg(v)}\) is an even number. Because \(\sum_{v \in Even \ deg(v)}\) is even, \(\sum_{v \in Odd \ deg(v)}\) must be even as well. For this to be true, the number of vertices with odd degree must be even, q.e.d.
3.2 Fact 2: Weight of minimal matching

Claim 3.2.1 Let \( G \) be a complete graph satisfying the triangle inequality and let \( G' \) be a subgraph of \( G \) such that \( G' \) is complete and has an even number of vertices. Then the weight of a minimal matching on \( G' \) must be at most \( \frac{OPT}{2} \), i.e. at most half of the weight of the minimum weight Hamiltonian cycle on \( G \).

Proof: First note that the weight of the minimum weight Hamiltonian cycle on \( G' \) is no larger than that on \( G \). This is because we can take the minimum weight Hamiltonian cycle computed for \( G \) and remove from it any vertex not in \( G' \). We can replace the pair of edges connected to the vertex being removed with a single edge that connects the other two endpoints of the original two edges. The triangle inequality guarantees that the new edge weights no more than the sum of those it replaces. By iteratively removing vertices in this manner, we will eventually get to a Hamiltonian cycle on \( G' \), whose weight is at most that of the minimum weight Hamiltonian cycle on \( G \). It follows that \( OPT_{G'} \leq OPT_G \).

Since the number of vertices in \( G' \) is even, the minimum weight Hamiltonian cycle corresponding to it can be divided into two alternating paths that are both matchings of \( G' \). The lighter of these two matchings \( M_1 \) has at most weight \( \frac{OPT_{G'}}{2} \), where \( OPT_{G'} \) is the weight of the whole cycle. The minimal matching must therefore also have weight of at most \( \frac{OPT_{G'}}{2} \leq \frac{OPT_G}{2} \), q.e.d.

3.3 Fact 3: Minimum weight matching complexity

Claim 3.3.1 Given a complete weighted graph with an even number of vertices, a minimal weight perfect matching can be found in polynomial time.

For the proof see Paths, Trees, and Flowers by Jack Edmonds [3]. The proof goes beyond the scope of this class.

The running time of even bad implementations of Edmonds’ algorithm is certainly bounded by \( O(n^4) \) times a polylogarithmic factor. This dominates
the running time of the other steps of Christofides algorithm, resulting in Christofides algorithm also running in $O(n^4)$.

### 3.4 Fact 4: Existence of Eulerian Tour

**Claim 3.4.1** Any connected graph whose vertices are all of even degree has an Eulerian Tour.

We do not prove this here. The reader is referred instead to the textbook for Math61. It is easy to see why the converse is true, i.e. that odd degree vertices create problems, because every time you enter and exit a vertex along new edges, that uses an even number of edges. To prove that this is the only case where there are problems, involves an inductive argument that merges a maximum tour so far with a new cycle.

### 4 Food for Thought

1. Is $\frac{3}{2}$ a tight analysis of Christofides algorithm? Perhaps its worst case is actually better than $\frac{3}{2}$ but we have failed to analyze it correctly. *Challenge:* Find an example input for which Christofides algorithm will generate a solution that is exactly $\frac{3}{2}$ times optimal.

2. Is $\frac{3}{2}$ the best that we can do in polynomial time without having to prove P = NP? If a lower bound does exist, what is it?

### 5 Euclidean TSP

There is a special case of Metric TSP called Euclidean TSP. In Euclidean TSP, the weight of edges corresponds to ordinary (Euclidean) distances between their endpoints in the plane. Unfortunately, this problem is also NP-hard.

Vaidya has found an algorithm to find a minimum weight matching in Euclidean TSP that is $O(n^2 \log^4(n))$ [4]. This causes the overall running
time of Christofides algorithm to come down to the same asymptote in the case of Euclidean TSP.

Sanjeev Arora has found a Polynomial Time Approximation Scheme (PTAS) for Euclidean TSP [1].

References


