CS 787: Advanced Algorithms and Data Structures 11/03/03

Lecture 27: Christofides' Algorithm

Instructor: Eric Bach Scribe: Ye Zheng

In the last lecture, we discussed an approximation algorithm for TSP with $\alpha = 2$. Today we will improve the ratio from 2 to 3/2. In order to achieve this, we will use a more sophisticated algorithm and some results from the graph theory.

1 Euler Tours

The Königsberg bridge problem asks if the seven bridges of the city of Königsberg (formerly in Germany but now known as Kaliningrad and part of Russia) over the river Preger can all be traversed in a single trip without doubling back. In addition, we require that the trip ends in the same place it began. See Figure 1 for a depiction of the problem.

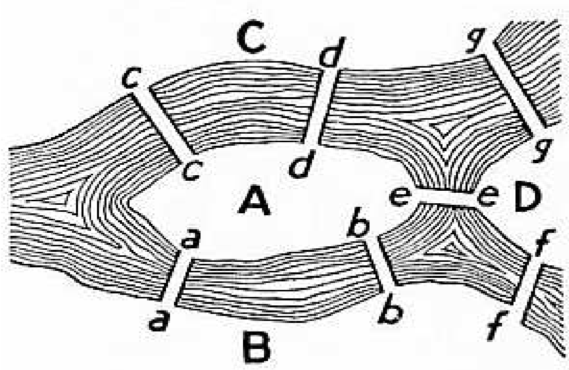

Figure 1: The Königsberg Seven Bridges by Kraitchik 1942 [MR]

Euler solved this problem (1736) by showing that it was impossible to satisfy the requirements. His study represented the beginning of graph theory. This problem is equivalent to asking if the multi-graph on four nodes and seven edges (Figure 2) has an Eulerian circuit (Euler tour).

Definiton (Euler Tour / Eulerian Circuit). Given a graph $G = (V, E)$, An *Euler tour* of G starts and ends at the same graph vertex and visits all vertices. In other words, it is a graph cycle where each graph edge is used exactly once. The term *Eulerian circuit* is also used synonymously with Euler tour.

For technical reasons, Eulerian circuits are mathematically easier to study than are Hamiltonian circuits. As a generalization of the Königsberg bridge problem, Euler showed (without proof) that a connected graph has an Eulerian circuit if and only if it has no graph vertices of odd degree.

Theorem 1. *A graph $G = (V, E)$ has an Euler tour if and only if it is connected and each vertex has even degree.*

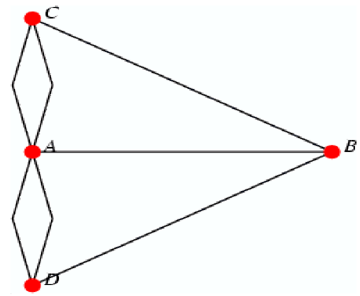

Figure 2: Abstraction of Königsberg bridge problem

Fleury's algorithm is an elegant method of generating Eulerian circuit.

Algorithm 1: Fleury's Algorithm for Euler Tour
Input: A connected graph $G = (V, E)$
Output: An Eulerian circuit of G
(1) Start at any vertex $v \in V$
(2) **while** untraversed edges remain
(3) Select a new edge $(v, w) \in E$ (avoiding cut edge unless unavoidable)
(4) Output (v, w)
(5) $v \leftarrow w$

Note that a *cut edge* is an edge whose removal disconnects the graph. If we remove the cut edge and there still exists a non-cut edge (say e). We cannot go back to the component that contains e. In addition, we can use the depth first search to decide whether an edge is a cut-edge or not. So Fleury's algorithm is polynomial time bounded.

2 Matchings

Definiton (Matching). A *matching* on a graph $G = (E, V)$ is a set of edges of G such that no two of them share a vertex in common. The largest possible matching consists of half of the edges, and such a matching is called a *perfect matching*.

Note that although not all graphs have perfect matchings, a maximum matching exists for each graph. There exists a polynomial time algorithm to find a minimum cost perfect matching in a complete weighted graph (also called the assignment problem). [1] Now we are ready to introduce the 3/2-approximation algorithm.

3 Christofides' Algorithm

Recall that the TSP asks that given a complete undirected graph $G(V, E)$ that has a non-negative integer cost $c(u, v)$ associated with each edge $(u, v) \in E$, find a path which starts and ends at the

[1] The assignment problem is the maximum weight matching problem in the bipartite graph.

same vertex (a tour), includes every other vertex exactly once, and minimizes cost. We focus on the cost function which satisfies the triangle inequality. The cost function c satisfies the triangle inequality if for all vertices $u, v, w \in V$,

$$c(u, w) \leq c(u, v) + c(v, w)$$

In other words, the cheapest (shortest) way of going from one city to another is the direct route (i.e., straight line) between two cities. In particular, if every city corresponds to a point in Euclidean space, and distance between cities corresponds to Euclidean distance, then the triangle inequality is satisfied.

Algorithm 2: Christofides Algorithm
Input: a weighted graph $G = (E, V)$
Output: a TSP tour
(1) Find a minimum spanning tree T of G
(2) Let G' be subgraph of G induced by vertices of odd degree in T. Then G' has even number of vertices.
(3) G' =complete graph with even number of vertices. Let M be a minimum cost perfect matching of G'
(4) $T + M$ (multiple edges included) has a Eulerian circuit K by Theorem 1
(5) Take shortcuts in K to get a TSP tour.

We illustrate the algorithm by the example in Figure 3 [HC].

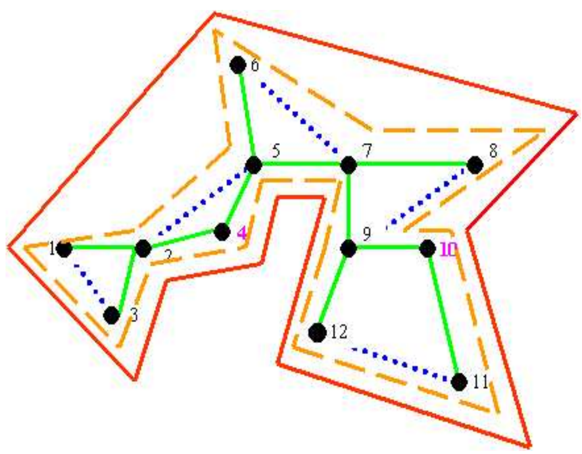

Figure 3: Christofides' Algorithm

In step 1, we find the Minimum Spanning Tree T_1 (the green edges). The sum of degrees of all the vertices $S(d) = 2m$, where m is the number of edges. Therefore $S(d)$ is even. Let $S_e(d)$ to be the sum of degrees of the vertices which have even degree, $S_e(d)$ is also even. Therefore $S(d) - S_e(d) = 2k$, $k = 1, 2, \ldots$ which means that the sum of degrees of the vertices which have odd degree is also an even number. Thus there are even numbers of vertices which have odd degree. In the example, vertices 4 and 10 have even number degrees. Thus the set of vertices with odd degree is $G' = \{1, 2, 3, 5, 6, 7, 8, 9, 11, 12\}$.

In step 3, we find a minimum weight matching on the vertices in G'. In this example, the minimum matching edges are 1-3,2-5,6-7,8-9,11-12. (see the blue dashed lines in the Figure 3)

In order to prove the approximation ratio is 3/2, we need to use the following lemma:

Lemma 1. *$G = (V, E)$ is a graph. Let $W \subseteq V$ with even cardinality $|W|$, and M be a minimum cut matching for the subgraph induced by W. Then*

$$\text{cost}(W) \leq \text{OPT}/2$$

Proof. Take any optimal tour of G and take the shortcuts to make a tour K for W. $K = M \cup M'$ has even length where M and M' are matchings for W.

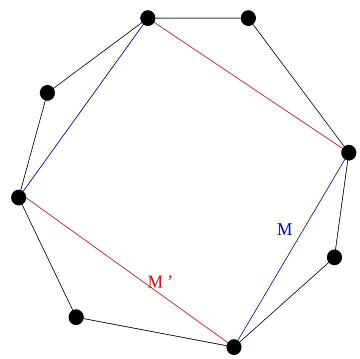

Figure 4: Proof of Correctness and Approximation Ratio

From the Figure 4, we see that

$$\begin{aligned} 2 \cdot \text{cost of the best matching for } W &\leq \text{cost}(M) + \text{cost}(M') \\ &= \text{cost}(K) \\ &\leq \text{cost of the optimal TSP tour} \end{aligned}$$

where the last inequality follows by the triangle inequality. Now observe that

$$\begin{aligned} 2 \cdot \text{cost of the Christofides tour} &\leq \text{cost of the Eulerian circuit} \\ &= \text{cost}(K) + \text{cost}(T) \\ &\leq \frac{1}{2}\text{OPT} + \text{OPT} = \frac{3}{2}\text{OPT} \end{aligned}$$

□

Christofides' algorithm is the best known approximation algorithm for the Euclidean TSP. We also can give an example to show that the 1.5 approximation ratio can be reached. The given graph G is shown in Figure 5.

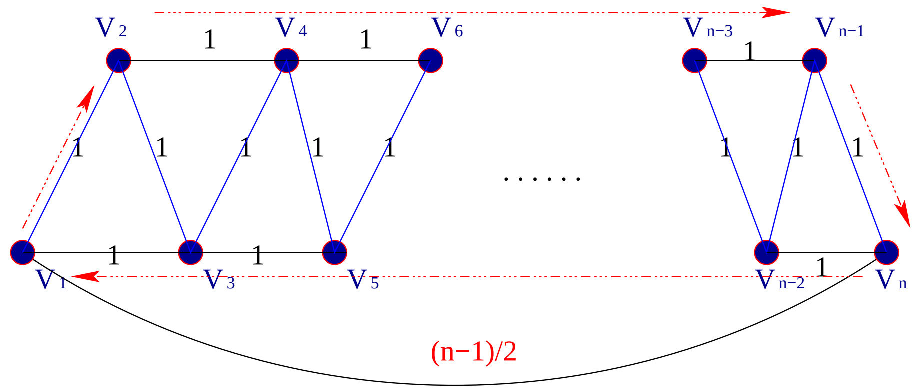

Figure 5: A bad case for Christofides' algorithm

The MST of G is the blue edges which cost $n-1$. So M consists of the edges with cost $(n-1)/2$. Therefore, Christofides' tour has cost $3n/2 + O(1)$ while the optimal tour has cost n (as shown in red dashed line).

References

[BM] A.V. Bondy and U.S.R. Murty. *Graph Theory with Applications*.

[MR] *Mathematical Recreations*, Dover, 1958.

[HC] `http://www.msci.memphis.edu/~giri/7713/f00/HuiChen/HuiChen2.htm#chris`.