Professor Lenore Cowen
Scribe: Stephanie Tauber

Lecture 3: The Travelling Salesman Problem (TSP)

1 Introduction

A salesman wishes to visit every city on his route exactly once (i.e., no path between two cities can be traversed twice) and return home, at minimal cost.

Input: A set of $n \geq 3$ cities. To each pair of cities (i, j) we associate a weight (distance in miles, cost, etc.) See figure 5.

Output: A closed path (i.e., starts and ends at the same vertex) or *tour* of minimum cost, where the *cost* of the tour is defined as the sum of the weights of the edges in the tour.

1.1 The Metric TSP

In this lecture, we consider a special case of TSP called Metric TSP. Metric TSP is a subcase of TSP where the Triangle Inequality holds. (This is always true when the weights on the edges are actual distances. See Figure 1.)

Definition 1.1 *The Triangle Inequality:*

$$\forall (i, j, k)\ d(i, j) \leq d(i, k) + d(k, j)$$

2 Approximation Algorithms for Metric TSP

Both TSP and Metric TSP are NP-hard problems, that is, there is no known polynomial-time algorithm for solving these problems, unless P=NP. But, we can use *approximation algorithms* to get within a certain factor of the optimal answer.

Let OPT denote the cost of the minimum weight tour:

Goal 1: A polynomial-time algorithm that outputs a tour of cost C, where $C \leq 2 \times OPT$.

Goal 2: Christofides' Algorithm: A polynomial-time algorithm that outputs a tour of cost C, where $C \leq 1.5 \times OPT$.

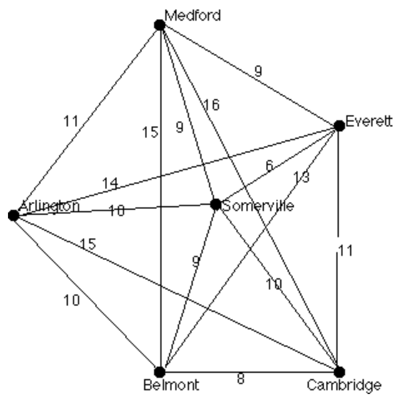

Figure 1: Metric TSP Example

2.1 Goal 1: $C \leq 2 \times OPT$

1. Take the minimum-weight spanning tree (MST) of the TSP graph. See Figure 2. The MST can be computed in polynomial time using, for example, Kruskal's or Prim's algorithm [1].

   Claim 2.1 *Weight of $MST \leq OPT$.*

   Proof By contradiction: Assume an instance of a Metric TSP with $OPT < MST$. Removing an edge from OPT creats an acyclic graph hitting every node once, therefore it is a spanning tree with weight $T < OPT$, so $T < OPT < MST$. Therefore MST was *not* minimal, a contradiction. ■

2. Do a depth-first search (DFS) of the MST, hitting every edge exactly twice. This "pseudotour" PT has cost=$2 \times MST \leq 2 \times OPT$. Write down each vertex as you visit it. See Figure 3.

3. To go from PT to T*: Rewrite the list of vertices, writing each vertex only the *first* time it appears in PT. In the example in Figure 3, the PT was ASMSESBCBSA so T* is ASMEBC. See Figure 4.

   Claim 2.2 *Cost of $T* \leq PT$, therefore cost of $T* \leq 2 \times OPT$.*

   Proof Follows from the Triangle Inequality: Going from A to B must be cheaper than A to C to B. ■

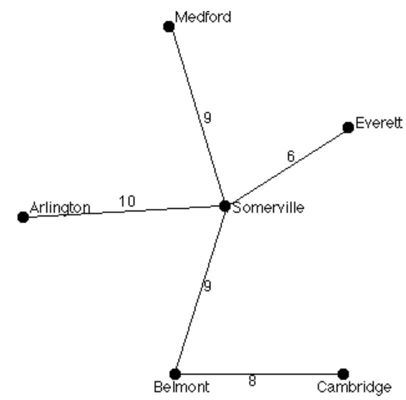

Figure 2: Minimum Spanning Tree of the TSP Graph

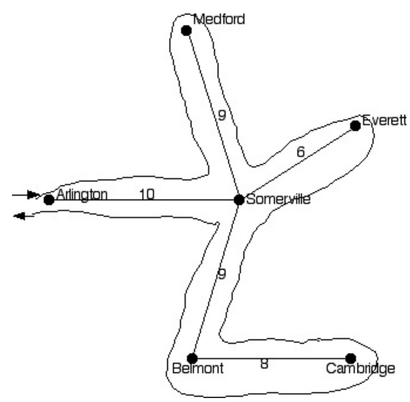

Figure 3: Depth-First Search Yielding Pseudotour ASMSESBCBSA

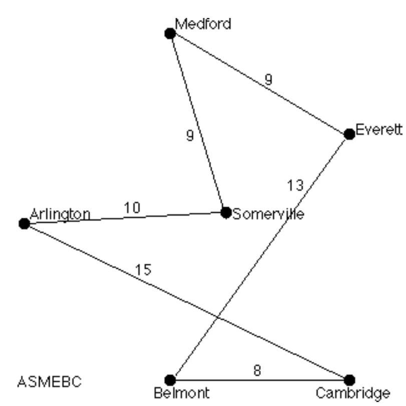

Figure 4: Resulting Tour T*=ASMEBC

2.2 Goal 2: $C \leq 1.5 \times OPT$ (Christofides' Algorithm)

Two facts we need:

1. Minimum-weight matching in a weighted complete graph can be found in polynomial time. [2]
2. A graph has a Eulerian tour if and only if all vertex degrees are even. In such a graph we can construct this tour in polynomial time.

   (A Eulerian path visits each edge exactly once and completes a circuit; a Eulerian tour is a Eulerian path that is also a tour, that is, it starts and ends in the same place.)

Christofides' Algorithm:

1. Create an MST, as before.

   Claim 2.3 *In any graph, the number of vertices of odd degree must be even.*

   Proof The sum of the degrees of all the vertices in a graph is equal to twice the number of edges, therefore it is an even number.

   $$\sum_{v \in V} deg(v) = 2 \times E$$

   Since an even and an odd number added together make an odd, it follows that the number of the degrees of the vertices of odd degree and the sum of the degrees of the vertices of even degree must both also be even.

   $$\sum_{v \in V} deg(v) = \sum_{v \in V_{odd}} deg(v) + \sum_{v \in V_{even}} deg(v)$$

   Therefore, since the sum of the odd degrees is an even number, there must be an even number of vertices of odd degree. ■

2. Find a minimum-weight perfect matching M* in the original graph between the vertices that have odd degree IN THE MST.

Claim 2.4 $M* \leq 0.5 \times OPT$

Proof Any tour can be decomposed into two matchings, M_1 and M_2, by alternating matched and unmatched edges. Therefore,

$$OPT = M_1 + M_2 \geq M * + M*$$

Consider the subgraph $MST + M*$: Every vertex in this subgraph has even degree, so there exists some Eulerian tour E of this subgraph with cost exactly equal to $MST + M*$ since it uses each edge exactly once. Therefore we have:

$$MST + M* \leq 0.5 \times OPT + OPT \leq 1.5 \times OPT$$

■

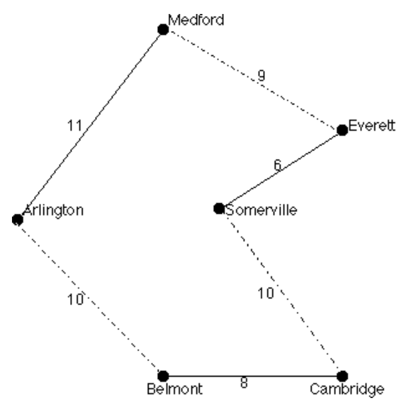

Figure 5: Resulting Tour E*

3. Repeat as above, write down each vertex the first time is appears in the Eulerian tour E, creating a salesman tour $E*$ with cost $\leq 1.5 \times OPT$. See Figure 5.

References

[1] T. Cormen, C. Leiserson, and R. Rivest, *Introduction to Algorithms*, The MIT Press, 1990.

[2] J. Edmonds, "Paths Trees and Flowers", *Canadian Journal of Math*, vol 17 (1965), 449-467.