Lecture 5: Approximation Algorithms for TSP\footnote{These notes were revised from scribe notes from Stephanie Tauber in 2002}

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 4.

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, Travelling Salesman Problem, called Metric TSP. Metric TSP is a subcase of TSP where the Triangle Inequality holds and distances are symetrical. (This is always true when the weights on the edges are actual distances. See Figure 1.)

**Definition 1.1.1** 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 2.1.1 Weight of MST $\leq OPT$.

Proof 2.1.2 By contradiction: Assume an instance of a Metric TSP with $OPT < MST$. Removing an edge from OPT creates 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 2.1.3 Cost of $T^* \leq PT$, therefore cost of $T^* \leq 2 \times OPT$.

Proof 2.1.4 Follows from the Triangle Inequality: Going from A to B must be cheaper than A to C to B.

References


Figure 2: Minimum Spanning Tree of the TSP Graph

Figure 3: Depth-First Search Yielding Pseudotour ASMSESBCBSA
Figure 4: Resulting Tour $T^\ast$ = ASMEBC