In this lecture we discussed:

- **Traveling Salesman Problem (TSP)**
- **Triangle TSP (Δ TSP)**
- **Algorithm for Δ TSP**

Handouts (posted on webpage):

- HANDOUT NAME: None

Reading: Probably want to check http://jeffe.cs.illinois.edu/teaching/algorithms/notes/31-approx.pdf.

## 1 Approximate Algorithms: Traveling Salesman Problem (TSP)

- **Traveling Salesman Problem (TSP)**
  
  definition: Given a graph with weight (distances) on its edges, you must find a cycle of minimum total weight that visits each vertex exactly once.

- **Δ TSP**
  
  This is similar to TSP except that Triangle TSP has edges weights obeying triangle inequality. The triangle inequality rule is illustrated in the following triangle

\[
\begin{array}{c}
\text{a} \\
\downarrow \\
\text{b} \\
\downarrow \\
\text{c}
\end{array}
\]

\[d(a, b) \leq d(a, c) + d(c, b)\text{ where } d\text{ is distance}\]

We will be discussing Δ TSP for the rest of this lecture.

- **Christofides Algorithm for triangle TSP (1976)**
  
  The algorithm is as below:
  
  1. Find a minimum spanning tree (MST) of G. call this T = MST
Figure 1: T (tree) matching to demonstrate TSP

2. Compute minimum length complete (perfect) matching $M$ in the complete graph on odd-degree vertices of $T$ (tree) $T$

3. Find Euler tour $E$ in $T \cup M$
   
   – we define Euler tour as a cycle that visits every edge exactly once.

4. Eliminate repeated vertices from $E$ to get TSP tour $R$

**Things to note**

1. A perfect matching always exists in step 2 above because there are an even number of odd degree vertices

2. Euler tour exist because every vertex in $T \cup M$ has even-degree.

3. A TSP tour minus one edge is a spanning tree: which implies that:
   
   $|T| \leq |TSP(G)|$ where $|T|$ is the total edge weight in $|T|

Figure 2: Two Odd-degrees matching; black as B and red as D

4. – figure 2 above shows two matching for odd degree vertices
call the black odd-degree matching B and the red odd-degree matching D then we have

- \(|B| + |D| \leq |TSP(G)|\) therefore,
- \(|M| \leq |B|\) and \(|M| \leq |D|\) because M is minimum weight matching which implies that: \(|M| \leq \frac{1}{2}|TSP(G)|\)
- Euler tour E has \(|E| \leq \frac{3}{2}|TSP(G)|\)
- And the final approximation tour is: \(|R| \leq |E| \leq \frac{3}{2}|TSP(G)|\)

2 Euclidean TSP is NP-Hard (Papadimintrion 1977)

Hamiltonian Cycle

Given unweighted graph G, does G contain a cycle that visits every vertex once? That is the basis of Hamiltonian cycle.

What if we want to approximate the general TSP? Ans. No.

- **Hardness of Approximation**

The general TSP is NP-Hard to approximate

Claim: If \(P \neq NP\) then there is no polynomial time c-approximation algorithm for TSP

Proof

Suppose A is polynomial time c-approximation algorithm for TSP we use A to solve Hamiltonian cycle

Transform X: create G’ from G = (V,E).

Figure 3: Hamiltonian Cycle and Transformation
G’ has all edges
\[ w(u,v) = \begin{cases} 
1 & \text{if } (u,v) \in G \\
|V| + 1 & \text{if } (u,v) \not\in G
\end{cases} \]

Transform Y: \(|TSP_A(G')| \leq c|V|\) then output yes otherwise output no

**Why does it work?**

Edges not in the original graph are so costly that there is a gap between cost of tour if \(G\) contains a Hamiltonian cycle (\(cost = n\)) and cost of tour if \(G\) does not (\(cost > c|V|\))

3 **NEXT TOPIC**

- On-Line Algorithms