CS420+500: Advanced Algorithm Design and Analysis

Lectures: Mar 20 + Mar 22, 2017

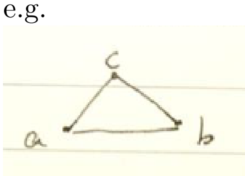
Prof. Will Evans *Scribe: Yuefan Liu*

In this lecture we discussed :

- Concepts (in Approximation TSP);
- Approximate algorithm for $\triangle$TSP;
- Euclidean TSP is NP-hard.

1 Concepts (in Approximation TSP)

- TSP (traveling salesman)
  Given: a graph with edge weights
  Return: visit every vertex (and return home) exactly once, using smallest total weight
- $\triangle$TSP (Triangle TSP)
  Edge weight obey triangle inequality
  i.e. $a \to c \to b$ cost more than $a \to b$
  e.g.

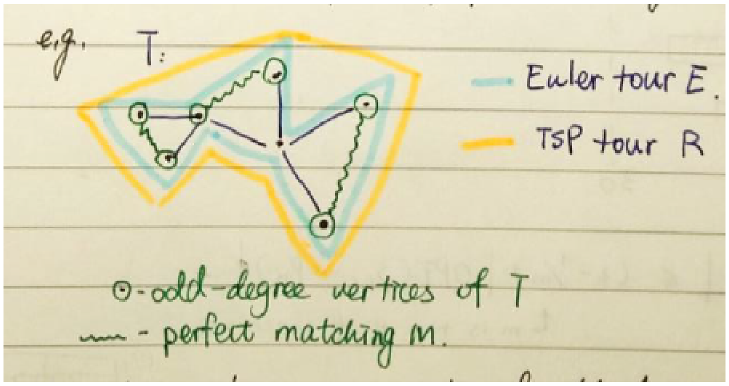


  triangle inequality: $d(a,b) \le d(a,c) + d(c,b)$

2 Christofides algorithm for $\triangle$TSP (1976)

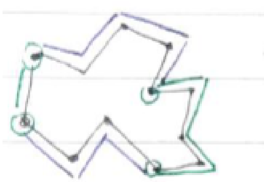
1. Find T = minimum spanning tree of G
2. Compute minimum length complete (perfect) matching M in the complete graph on odd-degree vertices of T
3. Find Euler tour E in T $\cup$ M (a Euler cycle exists when all vertices are even degree in the graph, and the Euler cycle visit every edge exactly once)
4. Eliminate repeated vertices from E to get TSP tour R

   Image:



Note:

- A perfect matching always exists in step 2 because there are an even number of odd-degree vertices (remind that the total degree of all vertices in the graph is even)
- Euler tour exists because every vertex in T U M has even degree
- TSP tour minus one edge is a spanning tree, so $|T| \leq |TSP(G)| - - -$ ① , where T is the total edge weight
- This graph shows two matchings for odd degree vertices: one Blue B, one green D

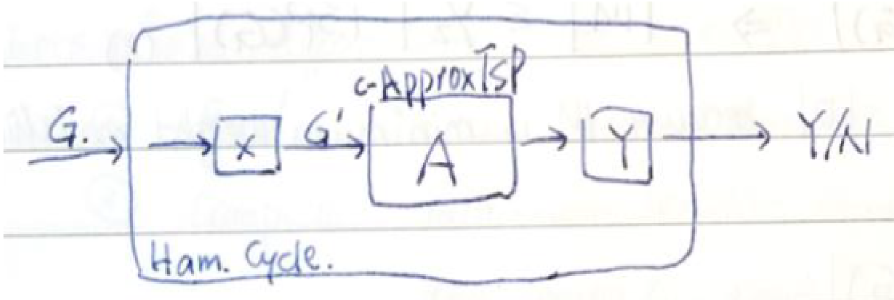
  

  Note that the cycle in the figure is a minimum tour. Therefore, the total weight of the edges of B and D is at most the minimum tour length
  $|B| + |D| \leq |TSP(G)| \Rightarrow |M| \leq \frac{1}{2}|TSP(G)| - - -$ ② , because $|M|$ is the minimum weight matching, $|M| \leq |B| and |M| \leq |D|$
- From ① and ②: Euler tour E has $|E| \leq 3/2|TSP(G)|$
- $|R| \leq |E| \leq \frac{3}{2}|TSP(G)|$

3 Euclidean TSP is NP-hard [Papadimitrion 77]

- Hamiltonian Cycle:
  Given: unweighted graph G
  Return: does G contain a cycle that visit every vertex once?
- Hardness of Approximation
  - What we know: 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 a polytime c-approx algorithm for TSP
    * We use A to solve Hamiltonian cycle

* Graph: X is in-put transformation; Y is output transformation



* Transform X:
  * Create $G\prime$ from $G = (V, E)$, $|V| = n$
  * $G\prime$ has all edges: $w(u, v) = 1((u, v) \in G), c|V| + 1((u, v) \notin G)$
* transform Y: if $|TSPA(G)| \leq c|V|$, then output YES otherwise NO

– Why does this 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 ($cost > c|V|$)