Algorithmic Workshop, Krakow	October 11, 2014

Lecture 1 (Symmetric TSP)

Lecturer: Ola Svensson *Scribes: Ola Svensson*

1 Introduction

- Welcome
- Ola Svensson (*ola.svensson@epfl.ch*)
- *http://theory.epfl.ch/osven*
- Goal of course: learn basic techniques and recent developments for the traveling salesman problem

2 Motivation: "Fast. Reliable. Cheap. Choose two."

Some desirable properties for an algorithm are:

1. runs in polynomial time,
2. finds exact optimal solution,
3. robust: works for any input instance of a problem.

Most natural optimization problems are NP-hard (e.g. set cover, chromatic number). We can find exact solutions using techniques from mathematical programming, for example. But we do not know how to do so with a guarantee of efficiency or polynomial running time. In this class, we study algorithms that satisfy the first and third properties—fast algorithms for every input—and so we must relax the second property.

2.1 What is an Approximation Algorithm?

An α-approximation algorithm A for a problem P is an algorithm that:

1. runs in polynomial time,
2. for any instance I of problem P, algorithm A produces a solution with value $val_A(I)$ such that:
   (a) $\frac{val_A(I)}{OPT(I)} \leq \alpha$ (if P is a minimization problem),
   (b) $\frac{val_A(I)}{OPT(I)} \geq \alpha$ (if P is a maximization problem).

3 Symmetric Traveling Salesman Problem

Given n cities with costs c_{ij} to go between city i and j, the problem is to find a minimum cost tour that visits each city at least once. We can think of this problem on a graph, $G = (V, E)$, where the n cities are represented by vertices in V and each edge has weight c_{ij}. We can assume the triangle inequality: for all $i, j, j \in V$, $c_{ij} \leq c_{ik} + c_{kj}$, and with this assumption, our goal is to find the minimum cost tour that visits each vertex exactly once. We will refer to this problem as the *metric traveling salesman problem* or *metric TSP*.

Metric-TSP

Given: A (complete) graph $G = (V, E)$ with nonnegative edge costs $\{c_e\}_{e \in E}$ satisfying the triangle inequality.

Find: A connected Eulerian graph $G = (V, E')$ where $E' \subseteq E$ may contain several copies of the same edge so as to minimize $\sum_{e \in E'} c_e$.

An Eulerian graph is a graph where each vertex has even degree.

4 Double Spanning Tree Algorithm

What is a polynomial-time computable lower bound on the cost of an optimal solution for the metric TSP problem?

Lemma 1 *The cost of a minimum spanning tree is at most that of an optimal tour.*

Proof Consider an optimal tour. Removing one edge results in a spanning tree with cost at most that of the tour. Since the minimum spanning tree has cost at most the cost of any tree, it follows that the cost of the minimum spanning tree is at most the cost of an optimal tour. ■

Algorithm "Double"
Input: A $G = (V, E)$ with edge costs $\{c_{ij}\}$.

1. Find a minimum spanning tree of G with cost $c(MST)$.
2. Double each edge in the spanning tree.
3. Take an Eulerian tour T of the doubled spanning tree.
4. Delete each previously visited vertex in the Eulerian tour T.

Output: Tour of the vertices, T.

First step ensures connectivity; second step ensures even degrees.

Note that in Step 2. of Algorithm "Double", we obtain a graph in which each vertex has even degree. It is well known that such a graph has an **Eulerian tour**, i.e. a tour that visits each edge exactly once, and there are many efficient ways to compute such a tour (it is iff). The resulting tour can be viewed as an ordering of the vertices in V in which each vertex appears at least once but may appear multiple times. The last step of Algorithm "Double" (Step 4) is to go through the vertices in the order determined by the Eulerian tour and delete an occurence of a vertex if it appeared previously in the order. Note that this step does not increase the cost of the tour due to the assumed triangle inquality on the edge costs.

Lemma 2 *Algorithm "Double" is a 2-approximation algorithm for metric TSP.*

Proof Call the cost of the final output tour of the vertices $c(T)$. Since Step 3 produces a tour that covers each vertex at least once, and Step 4 removes only repeated occurences of a vertex, the output tour T is a feasible solution. Moreover, the Eulerian tour computed at Step 3 has cost at most $2 \cdot c(MST)$. As mentioned earlier, due to the triangle inequality, we do not increase the cost of the tour in Step 4. Thus, for the cost of the output tour T, applying Lemma 1 we have:

$$c(T) \leq 2 \cdot c(MST) \leq 2 \cdot OPT.$$

■

5 Christofides' Algorithm

We are now ready to present Christofides' Algorithm:

ALGORITHM "CHRISTOFIDES"
Input: A complete graph $G = (V, E)$ with edge costs $\{c_{ij}\}$ satisfying the triangle inequality.

1. Find a minimum spanning tree MST of G.
2. Let O be the odd degree vertices in MST; find a minimum perfect matching PM of the vertices in O
3. Take an Eulerian tour T of the graph with edge set $MST \cup PM$.
4. Delete each previously visited vertex in the Eulerian tour T.

Output: Tour of the vertices, T.

again the first step is to ensure connectivity; second step is to ensure that all vertices have even degree

Exercise 1 *Show that any graph has an even number of odd-degree vertices.*

Lemma 3 *Algorithm "Christofides" is a* 1.5*-approximation algorithm for metric TSP.*

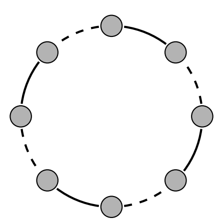
Proof As mentioned previously, we do not decrease the length of the tour by shortcutting. Therefore the cost of the returned tour is at most

$$c(MST) + c(PM)$$

We now that $c(MST)$ is at most OPT and we therefore finish the proof with the following claim.

Claim 4 *We have that* $c(PM) \leq OPT/2$.

Proof Consider the odd-degree vertices O. And consider the optimal tour restricted to these vertices. By the triangle inequality, this restricted tour has cost at most OPT. Now color edges alternatively by blue and red. Note that the blue edges form a perfect matching and the red edges form a perfect matching. As these edges partition the restricted tour, we have that one of them costs at most $OPT/2$.



■

The above claim finished the analysis of Christofides' algorithm. ■

Christofides' algorithm dates to 1976 and still remains the best. There is a famous conjecture that there exists an efficient algorithm that produces a tour of cost at most 4/3-optimal.

6 Graph TSP

In light of the apparent difficulty in addressing the metric TSP problem, researchers have considered the following special case called Graph TSP

Graph-TSP

Given: A graph $G = (V, E)$ with unit edge costs.

Find: A connected Eulerian graph $G = (V, E')$ where $E' \subseteq E$ may contain several copies of the same edge so as to minimize $\sum_{e \in E'} c_e = |E'|$.

Note that the above definition is equivalent to the metric-TSP except that all edges have the same cost. One can see that this is equivalent to the metric-TSP on shortest path metrics of unweighted graphs. The motivation for studying graph-TSP stems from the fact that it captures many of the difficulties of metric-TSP but at the same time, it is easier to argue about.

6.1 Removable Pairing Algorithm

We shall explain the main ideas behind the techniques by proving the following result

Lemma 5 *Any 2-edge connected cubic graph $G = (V, E)$ has a tour of length $4n/3 - 2/3$.*

A graph is cubic if every vertex has degree 3 and it is 2-edge connected if one needs to delete at least 2 edges to make it disconnected.

Exercise 2 *Any 2-edge connected subcubic graph $G = (V, E)$ has a tour of length $4n/3 - 2/3$.*

This result is tight as can be seen by considering 3 long paths between two vertices s and t.

Before proving the above lemma, we shall need the following:

Lemma 6 *Any 2-edge connected cubic graph $G = (V, E)$ has a distribution μ of perfect matchings such that*

$$\Pr_{M \sim \mu}[e \in M] = 1/3 \qquad \forall e \in E.$$

Proof Edmonds perfect matching polytope has the following description

$$\begin{aligned} x(\delta(v)) &= 1 & \forall v \in V \\ x(\delta(S)) &\geq 1 & \forall S \subseteq V \text{ of odd cardinality} \end{aligned}$$

Exercise 3 *For a 2-edge connected cubic graph, $x_e = 1/3$ for all $e \in E$ is a feasible solution to Edmond's perfect matching polytope.*

Therefore it follows that $x_e = 1/3$ for all $e \in E$ can be written as a convex combinations of perfect matchings, i.e.,

$$x = \sum_{i=1}^{k} \lambda_i y^{(i)}, \qquad \lambda_i \geq 0, \text{ and } \qquad \sum \lambda_i = 1.$$

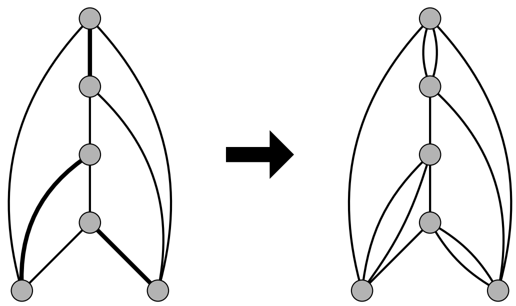
This gives our distribution. ■

6.2 Algorithm 1st trial

Algorithm "1st Trial"
Input: A 2-edge connected cubic graph $G = (V, E)$.

1. Sample a perfect matching M from μ

Output: Output the Eulerian connected graph $E \cup M$.



Lemma 7 *The expected cost of the returned solution is at most $2n$. (It is actually equal to $2n$ in this case.)*

Proof

$$\begin{aligned}\mathbb{E}[|E \cup M|] &= \mathbb{E}[|E|] + \mathbb{E}[|M|] \\ &= |E| + |E|/3 \\ &= 3n/2 + n/2 = 2n\end{aligned}$$

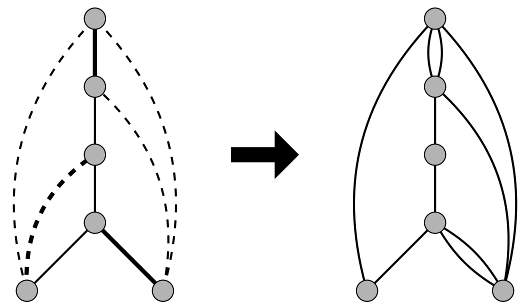
■

6.3 Algorithm 2nd trial

Algorithm "2nd Trial"
Input: A 2-edge connected cubic graph $G = (V, E)$.

1. Take a spanning tree T of G
2. Sample a perfect matching M from μ

Output: Output the Eulerian connected graph $E + (M \cap T) - (M \setminus T)$.



Lemma 8 *The expected cost of the returned solution is at most* $5n/3 - 2/3$.

Proof

$$\begin{aligned}\mathbb{E}[|E + |M \cap T| - |M \setminus T|] &= \mathbb{E}[|E|] + \mathbb{E}[|M \cap T|] - \mathbb{E}[|M \setminus T|] \\ &= |E| + |T|/3 - 1/3(|E| - |T|) \\ &= 2|E|/3 + 2|T|/3 = n + 2(n-1)/3 = 5n/3 - 2/3\end{aligned}$$

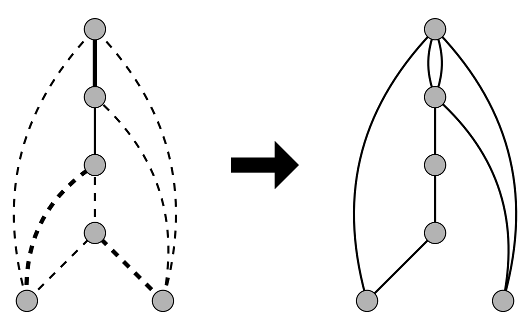
■

6.4 Algorithm 3rd trial

ALGORITHM "3RD TRIAL"
Input: A 2-edge connected cubic graph $G = (V, E)$.

1. Obtain a spanning tree T of G by depth-first search and direct edges in T from the root and back-edges toward the root
2. Let R be the back-edges plus any tree (u, v) if u has an incoming back-edge
2. Sample a perfect matching M from μ

Output: Output the Eulerian connected graph $E + (M \setminus R) - (M \cap R)$.



Lemma 9 *The output is always a connected graph.*

Proof We prove it by simple induction on the depth of the tree. In the base case, the graph is just a vertex so trivially true.

In the inductive step consider a vertex with $\ell \in \{1,2,3\}$ children $w_1, \ldots, w_\ell$ in T. By I.H the trees rooted at w_j stay connected. It is thus enough to verify that v stays connected to each of these trees. If (v, w_j) is not in R then it clearly holds. If (v, w_j) is in R then by definition exists a back edge (u, v) where u belongs to the tree rooted at w_j. In a perfect matching only one of these edges can be deleted so we are good. ■

Lemma 10 *The expected cost of the returned solution is at most* $4n/3 - 2/3$.

Proof

$$\begin{aligned}\mathbb{E}[|E + |M \setminus R| - |M \cap R|] &= \mathbb{E}[|E|] + \mathbb{E}[|M \setminus R|] - \mathbb{E}[|M \cap R|] \\ &= |E| + (|E| - |R|)/3 - |R|/3 \\ &= 4|E|/3 - 2|R|/3.\end{aligned}$$

Now the cardinality of R is $2 \cdot (|E| - (n-1)) - 1$ so we have

$$\begin{aligned}4|E|/3 - 2|R|/2 &= 4|E|/3 - 4/3(|E| - (n-1)) + 2/3 \\ &= \frac{4}{3}(n-1) + 2/3 = 4n/3 - 2/3\end{aligned}$$

■