# Approximating Metric TSP

Shourya Pandey

April 13, 2019

# Overview

# Introduction

# Introduction

- In this presentation, we will describe the Travelling Salesman Problem (TSP), and we will prove that it is hard to find approximate solutions for TSP.
- We will then talk about the **Metric TSP** problem, prove its hardness, and provide approximation algorithms for it.

# The Hamiltonian Cycle Problem (HC)

- A **Hamiltonian cycle** (or a Hamilton cycle) on a graph $G(V, E)$ on $n$ vertices is a cycle that visits each vertex exactly once, that is, the cycle has length $n$.

# The Hamiltonian Cycle Problem (HC)

- The Hamiltonian Cycle Problem can be stated as follows:
  - Input: A graph $G(V, E)$.
  - Output: Does there exist a Hamiltonian Cycle in $G$?
- It is known that **HC is NP-hard**.

# The Travelling Salesman Problem (TSP)

- The **Travelling Salesman Problem (TSP)** is, in some ways, an extension of the HC Problem, and can be stated as follows:
  - Input: A graph $G(V, E)$ with a non-negative cost function $c$ on the edges.
  - Output: A Hamiltonian cycle of minimum cost.
- An easy reduction shows that this problem is at least as hard as the previous problem.

# The Travelling Salesman Problem (TSP)

- It is not hard to see that we can reformulate the problem as follows:
  - Input: A **complete** graph $G(V, E)$ with a non-negative cost function $c$ on the edges.
  - Output: A Hamiltonian Cycle of minimum cost.
- **TSP is NP-hard**.

# Can we approximate TSP?

- We do not hope to solve TSP in polynomial time. Can we find an $\alpha$-approximation for TSP in polynomial time?
- Such problems are called **APX** problems.
- In complexity theory the class APX (an abbreviation of approximable) is the set of NP-optimization problems that allow polynomial-time approximation algorithms with approximation ratio bounded by a constant.
- It turns out that we cannot even approximate TSP.

# Hardness of Approximation of TSP

**Claim:** For every $\alpha > 1$, there exists no polynomial time algorithm approximating TSP within a factor of $\alpha$, unless P $=$ NP.

# Hardness of Approximation of TSP

**Proof:** Assume the contrary, that is, suppose that for some $\alpha > 1$, we have a polynomial time algorithm $A$ that gives us an $\alpha$-approximation for TSP. We show a reduction from HC.

- We are given a graph $G(V, E)$, and we want to decide whether $G$ has a Hamiltonian Cycle.
- Construct a complete graph $G'$ on the same vertex set $V$, and allot costs to the edges of $G'$ as follows. For each edge $e$ in $G$:
  - If $e \in G$, give it cost $c(e) = 1$.
  - If $e \notin G$, give it cost $c(e) = \alpha \cdot n + 1$.
- It is clear that $G'$ is constructible in polynomial time.

# Hardness of Approximation of TSP

- Suppose $G$ has a Hamiltonian cycle $C$. Then in $G'$, the same cycle has a cost of $n$, and this is indeed the smallest possible cost (because each edge in $G'$ has cost at least 1).
- Therefore, the algorithm $A$, when run on $G'$, outputs an answer $\leq \alpha \cdot n$ in polynomial time.

# Hardness of Approximation of TSP

- Suppose $G$ does not have a Hamiltonian cycle. Then in $G'$, any Hamiltonian cycle must use at least one edge which is not in $G$, so any Hamiltonian cycle in $G$ has cost at least $\alpha \cdot n + 1$.
- Therefore, the algorithm $A$, when run on $G'$, outputs an answer $\geq \alpha \cdot n + 1$ in polynomial time.

# Hardness of Approximation of TSP

- In all, given $G$, construct $G'$ and run $A$ on $G'$.
- If $A$ outputs a number $\leq \alpha \cdot n$, output "yes", else output "no".

Therefore, we now have a polynomial time algorithm for HC, which is an NP-complete problem. Therefore, unless P = NP, TSP cannot have an $\alpha$-approximation algorithm that runs in polynomial time, for any $\alpha > 1$.

# Special TSPs

- Often, the graphs with edge costs that we encounter have other special properties associated with them. For example, the vertices may be houses that an actual salesman has to visit, and the costs represent the distances between houses.
- Such an instance of TSP is called **Euclidean TSP**, because the edges represent Euclidean distances between the vertices embedded in a plane.
- We can use other metrics to define costs as well.

# Metric TSP

# Metric

- A **metric** $d$ on a set $X$, also called a **distance function**, is a function that defines a distance between each pair of elements of the set. A set with a metric is called a **metric space**.
- Formally, $d : X \times X \to \mathbb{R}$ is a metric if it is a function satisfying the following properties for all $x, y, z \in X$:
  - *Non-negativity* : $d(x, y) \geq 0$
  - *Indiscernability* : $d(x, y) = 0$ iff $x = y$
  - *Symmetry* : $d(x, y) = d(y, x)$
  - *Subadditivity* : $d(x, y) + d(y, z) \geq d(x, z)$

# Metric

- **Example 1 :** The discrete metric, defined as $d(x, x) = 0$ and $d(x, y) = 1$ for all $x \neq y$.
- **Example 2 :** The Euclidean metric. The vertices represent actual points in some Euclidean space.
- **Example 3 :** The Graphic metric, where distance between two vertices is equal to the distance between them on a fixed tree.

# Metric TSP

- The **Metric TSP** problem is similar to the TSP problem, except now, the costs on the edges satisfy the triangle inequality.
  - Input: A complete graph $G(V, E)$, and a metric $c$ on $V$.
  - Output: A Hamiltonian cycle of minimum cost.
- How hard is Metric TSP?

# Hardness of Metric TSP

**Claim:** Metric TSP is NP-hard.
**Proof:** This is also a simple reduction problem from HC. Suppose we are given a graph $G(V, E)$ and we want to decide if it has a Hamiltonian cycle. Construct a complete graph $G'$ on the vertex set $V$, with cost function $c$ defined as follows:

- If $e \in G$, assign $c(e) = 1$ in $G'$.
- If $e \notin G$, assign $c(e) = 2$ in $G'$.

Clearly, $c$ is a metric on $V$. It is not hard to see that $G$ has a Hamiltonian cycle if and only if $G'$ has a Hamiltonian cycle of cost $n$.
This reduction shows that Metric TSP is NP-hard. By the way, such metric TSPs in which all weights are either 1 or 2 are called **(1,2)-TSPs**.

# Approximation Algorithms for Metric TSP

# A 2-approximation for MTSP

From now on, we refer to Metric TSP as **MTSP**. We first give a simple, 2-approximation for MTSP.

# A 2-approximation for MTSP

**2-approximation for MTSP:**

- We have a complete graph $G(V, E)$ with a metric $c$ on $V = \{1, 2, 3, \cdots, n\}$. Since $G$ is connected, we can find a **Minimum Spanning Tree (MST)** in polynomial time using, say, Prim's Algorithm. Call this MST T.

# A 2-approximation for MTSP

- Also, let $C^*$ be the cost of the smallest Hamiltonian cycle in $G$. Take some edge $e \in C^*$. Then $C^* \setminus e$ is also a spanning tree of $G$, so

$$c(T) \leq c(C^* \setminus e) \leq c(C^*)$$

# A 2-approximation for MTSP

- Root the MST at some vertex, say 1. Perform an Euler Tour on $T$ (which is the same as finding an Eulerian Circuit in the graph $T$ with all edges duplicated), and list the vertices that are visited in the order they are visited. Call this list $L$, and its cost is defined as the sum of the cost of the edges in $L$ (with repetition).

# A 2-approximation for MTSP

- Retain only the first occurrence of each vertex in this list, and also retain the last vertex in the list, which is 1, in $G$. This list can be interpreted as a cycle $C$ in $G$, because each vertex occurs exactly once in the list except for the first vertex, which also occurs in the end.
- This procedure will be referred to as **short – circuiting**.

# A 2-approximation for MTSP

- We claim that $c(C) \leq 2c(C^*)$.
- In an Euler Tour, each edge of $T$ is visited exactly twice, so $c(L) = 2c(T) \leq 2c(C^*)$.

# A 2-approximation for MTSP

- Also, $C$ has been formed by deleting vertices from the list $L$. Suppose

$$L = 1 \cdots ivj \cdots 1$$

and we delete this $v$ from $L$ to get the list $L'$. Then

$$c(L') = c(L) - c(iv) - c(vj) + c(ij) \leq c(L)$$

by triangle inequality.

- In all, this means $c(C) \leq c(L) \leq 2c(C^*)$, as required.

# Summarising the 2-Approximation Algorithm

**Algorithm:**

1. Create a minimum spanning tree $T$ of $G$.
2. Create a graph $H$ which is the graph $T$ but with all edges duplicated. Note that each vertex in $H$ now has even degree.
3. Find an Eulerian circuit $E$ in $H$.
4. Short-circuit $E$ to find the required Hamiltonian cycle $C$.

# Can we do better?

- The 2-approximation for MTSP was rather naive, in some sense. We can do better than this. What we really did was that we found a spanning tree $T$ and duplicated each edge in $T$ to get, say, $H$.
- This new graph $H$ has each vertex of even degree, so it has an Eulerian Circuit $E$.
- We then short-circuit $E$ to get a Hamiltonian cycle of cost at most twice the optimal cost.
- How can we improve this algorithm?
- Instead of duplicating each edge of $T$, we just need to ensure that the super-graph of $T$ has each edge of even degree. Nicos Christofides, in 1976, found a 1.5-approximation to Metric TSP by a small improvement in the algorithm.

# Christofides' Algorithm : A 1.5-approximation for MTSP

**1.5-approximation to Metric TSP:**

- As before, we have a complete graph $G(V, E)$ with a metric $c$ on $V$.
- Let $C^*$ be the optimal cycle in $G$, and let $T$ be an MST in $G$.

# Christofides' Algorithm : A 1.5-approximation for MTSP

- This is the crucial step. Let $O$ be the set of vertices that have an odd degree in $T$. Note that $|O|$ is even.
- Let $M$ be a minimum cost perfect matching in the subgraph induced by the vertices in $O$.
- Combine the edges of $M$ and $T$ to get a multigraph $H$.

# Christofides' Algorithm : A 1.5-approximation for MTSP

- Note that each vertex in $H$ now has even degree. Therefore, $H$ has an Eulerian Circuit $E$. $E$ can be found by any popular algorithm, say the Fleury's Algorithm or the Hierholzer's Algorithm.
- Make the cycle $C$ as done before by deleting repeated occurrences in $E$, that is, by short-circuiting.

# Christofides' Algorithm : A 1.5-approximation for MTSP

- Why did this work?
- We have

$$c(C) \leq c(E) \leq c(M) + c(T) \leq c(M) + c(C^*)$$

- We need to show that $c(M) \leq \frac{c(C^*)}{2}$. This is somewhat expected, seeing that $M$ has less than or equal to half the number of edges in $C^*$.

# Christofides' Algorithm : A 1.5-approximation for MTSP

- Consider the optimal cycle $C^*$. Short-circuit $C^*$ to create a cycle $C^O$ with the vertices of $O$ with a smaller cost than $c(C^*)$.
- Since $|O|$ is even, the edges in the cycle partition into two matchings $M_1$ and $M_2$. Suppose $M_1$ has a smaller cost than $M_2$.

# Christofides' Algorithm : A 1.5-approximation for MTSP

- Then,

$$c(M) \leq c(M_1) \leq \frac{c(M_1) + c(M_2)}{2} = \frac{c(C^O)}{2} \leq \frac{c(C^*)}{2}$$

  as required.
- Therefore $c(C) \leq 1.5 \cdot c(C^*)$.

# Summarising the Christofides' Algorithm

**Algorithm:**

1. Create a minimum spanning tree $T$ of $G$.
2. Let $O$ be the set of vertices in $T$ that have an odd degree. Find a minimum cost perfect matching $M$ on the subgraph induced by the vertices of $O$.
3. Create a graph $H$ which is the union of the edges of $T$ and $M$. This graph has all vertices of even degree.
4. Find an Eulerian circuit $E$ in $H$.
5. Short-circuit $E$ to find the required Hamiltonian cycle $C$.

# Running Time of the Christofides' Algorithm

- Creating an MST $T$ : $O(n^2)$
- Finding the minimum cost perfect matching $M$ in $O$ : $O(n^{2.5})$
- Creating $H$ : $O(n)$
- Finding an Eulerian circuit $E$ in $H$ : $O(n)$
- Short-circuiting $E$ : $O(n)$

The step that takes the longest time is the one to find the minimum cost perfect matching. The best known complexity as of now is $O(n^{2.5})$, given by Micali and Vazirani.

# A simple example which shows 1.5 is the best approximation this algorithm can give

# Conclusions

# Summary

- We described the HC and TSP Problems. We showed that it is hard to even approximate TSP.
- We then considered a special yet important class of TSPs called Metric TSPs. We proved that solving Metric TSP is also hard.
- We then discussed a naive 2-approximation to Metric TSP.
- Finally, we discussed the Christofides' Algorithm that gives a 1.5-approximation to Metric TSP.
- We also showed that 1.5 is the best approximation that the Christofides' Algorithm can give.

# What else is known about Metric TSPs?

- Christofides' Algorithm is the best known approximation algorithm for Metric TSPs as of now.
- It is known that Metric TSP cannot be approximated by a ratio better than $\frac{123}{122}$ unless P = NP. [Karpinski, Lampis, Schmied 2013]
- Similar results for many other special TSPs such as Graphic TSPs, (1,2)-TSPs, Cubic Graphic TSPs, etc. are known.

# The End