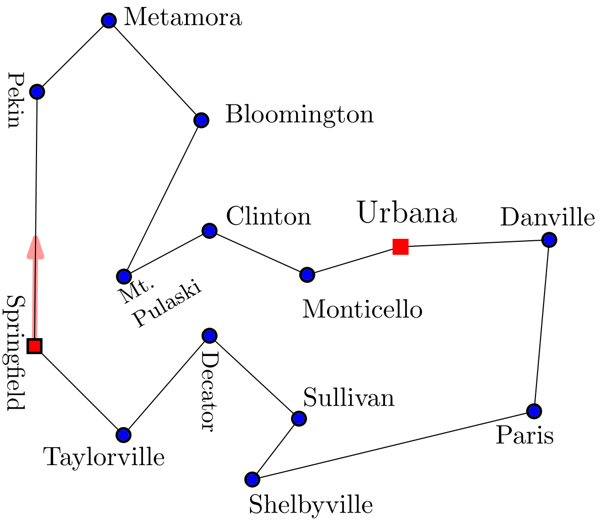
Approximation Algorithms for TSP

Lecture 26

May 1, 2018

Most slides are courtesy Prof. Chekuri

Lincoln's Circuit Court Tour



Traveling Salesman/Salesperson Problem (TSP)

Perhaps the most famous discrete optimization problem

Input: A graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$.
Goal: Find a Hamiltonian Cycle of minimum total edge cost

Traveling Salesman/Salesperson Problem (TSP)

Perhaps the most famous discrete optimization problem

Input: A graph $G = (V, E)$ with edge costs $c : E \rightarrow \mathbb{R}_+$.
Goal: Find a Hamiltonian Cycle of minimum total edge cost

Graph can be undirected or directed. Problem differs substantially.
We will first focus on undirected graphs.

Traveling Salesman/Salesperson Problem (TSP)

Perhaps the most famous discrete optimization problem

Input: A graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$.
Goal: Find a Hamiltonian Cycle of minimum total edge cost

Graph can be undirected or directed. Problem differs substantially.
We will first focus on undirected graphs.

Assumption for simplicity: Graph $G = (V, E)$ is a complete graph. Can add missing edges with infinite cost to make graph complete.

Traveling Salesman/Salesperson Problem (TSP)

Perhaps the most famous discrete optimization problem

Input: A graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$.
Goal: Find a Hamiltonian Cycle of minimum total edge cost

Graph can be undirected or directed. Problem differs substantially. We will first focus on undirected graphs.

Assumption for simplicity: Graph $G = (V, E)$ is a complete graph. Can add missing edges with infinite cost to make graph complete.

Observation: Once graph is complete there is always a Hamiltonian cycle but only Hamiltonian cycles of finite cost are Hamiltonian cycles in the original graph.

Inapproximability of TSP

Observation: In the general setting TSP does not admit any bounded approximation.

Inapproximability of TSP

Observation: In the general setting TSP does not admit any bounded approximation.

- Finding or even deciding whether a graph $G = (V, E)$ has Hamiltonian Cycle is NP-Hard

Inapproximability of TSP

Observation: In the general setting TSP does not admit any bounded approximation.

- Finding or even deciding whether a graph $G = (V, E)$ has Hamiltonian Cycle is NP-Hard
- **Hamiltonian Cycle $\leq_P$ (Approximate) TSP**

Inapproximability of TSP

Observation: In the general setting TSP does not admit any bounded approximation.

- Finding or even deciding whether a graph $G = (V, E)$ has Hamiltonian Cycle is NP-Hard
- **Hamiltonian Cycle $\leq_P$ (Approximate) TSP**
  - Suppose, $G = (V, E)$ is a simple graph in which we want to find a Hamiltonian cycle.
  - Construct a complete graph G' on vertices V, with cost 1 on edges of G and ∞ on all other edges.

Inapproximability of TSP

Observation: In the general setting TSP does not admit any bounded approximation.

- Finding or even deciding whether a graph $G = (V, E)$ has Hamiltonian Cycle is NP-Hard
- **Hamiltonian Cycle $\leq_P$ (Approximate) TSP**
  - Suppose, $G = (V, E)$ is a simple graph in which we want to find a Hamiltonian cycle.
  - Construct a complete graph G' on vertices V, with cost 1 on edges of G and ∞ on all other edges.
  - If G has a Hamiltonian cycle then there is a TSP tour of cost n in G', else the cost is ∞.

Important Special Cases

Metric-TSP: $G = (V, E)$ is a **complete graph** and c defines a metric space. $c(u, v) = c(v, u)$ for all u, v and $c(u, w) \leq c(u, v) + c(v, w)$ for all u, v, w.

Important Special Cases

Metric-TSP: $G = (V, E)$ is a **complete graph** and c defines a metric space. $c(u, v) = c(v, u)$ for all u, v and $c(u, w) \leq c(u, v) + c(v, w)$ for all u, v, w.

Geometric-TSP: V is a set of points in some Euclidean d-dimensional space $\mathbb{R}^d$ and the distance between points is defined by some norm such as standard Euclidean distance, L_1/Manhatta distance etc.

Metric-TSP

Metric-TSP is simpler and perhaps a more natural problem in some settings.

Theorem

Metric-TSP is NP-Hard.

Proof.

Metric-TSP

Metric-TSP is simpler and perhaps a more natural problem in some settings.

Theorem

Metric-TSP is NP-Hard.

Proof.

Given $G = (V, E)$ we create a new complete graph $G' = (V, E')$ with the following costs. If $e \in E$ cost $c(e) = 1$. If $e \in E' - E$ cost $c(e) = 2$.

Metric-TSP

Metric-TSP is simpler and perhaps a more natural problem in some settings.

Theorem

Metric-TSP is NP-Hard.

Proof.

Given $G = (V, E)$ we create a new complete graph $G' = (V, E')$ with the following costs. If $e \in E$ cost $c(e) = 1$. If $e \in E' - E$ cost $c(e) = 2$. Easy to verify that c satisfies metric properties. Moreover, G' has TSP tour of cost n iff G has a Hamiltonian Cycle. □

Metric-TSP: closed walk

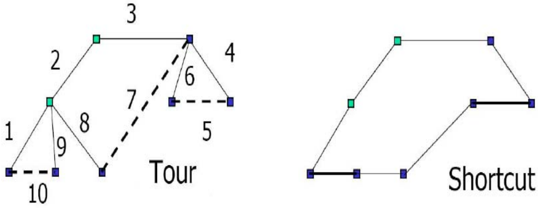
Metric-TSP: closed walk

Another interpretation of Metric-TSP: Given $G = (V, E)$ with edges costs c, find a tour of minimum cost that visits all vertices but can visit a vertex more than once – A closed walk.

Metric-TSP: closed walk

Another interpretation of Metric-TSP: Given $G = (V, E)$ with edges costs c, find a tour of minimum cost that visits all vertices but can visit a vertex more than once – A closed walk.

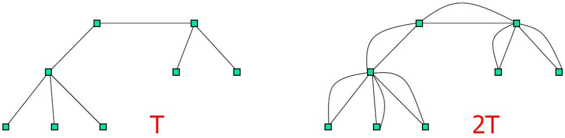
Because, any such tour can be converted in to a simple cycle of smaller cost by adding "short-cuts".



Approximation for Metric-TSP

MST-Heuristic$(G = (V, E), c)$

    Compute a minimum spanning tree (MST) T in G
    Obtain an Eulerian graph $H = 2T$ by doubling edges of T
    An Eulerian tour of H gives a tour of G
    Obtain Hamiltonian cycle by shortcutting the tour



Analyzing MST-Heuristic

Lemma

Let $c(T) = \sum_{e \in T} c(e)$ *be cost of MST. We have* $c(T) \leq OPT$*.*

Analyzing MST-Heuristic

Lemma

Let $c(T) = \sum_{e \in T} c(e)$ *be cost of MST. We have* $c(T) \leq OPT$.

Proof.

A TSP tour is a connected subgraph of G and MST is the cheapest connected subgraph of G. □

Analyzing MST-Heuristic

Lemma

Let $c(T) = \sum_{e \in T} c(e)$ *be cost of MST. We have* $c(T) \leq OPT$*.*

Proof.

A TSP tour is a connected subgraph of G and MST is the cheapest connected subgraph of G. □

Theorem

MST-Heuristic gives a 2*-approximation for Metric-TSP.*

Proof.

Cost of tour is at most $2c(T)$ and taking shortcuts only reduces the cost due to triangle ineuqlity. Hence MST-Heuristic gives a 2-approximation. □

Question

Consider the subgraph induced by edges of a tour that visits every vertex at least once (a closed walk). The degree of every vertex in this subgraph is:

1. Even
2. Odd
3. Either
4. Integer

Question

Consider the subgraph induced by edges of a tour that visits every vertex at least once (a closed walk). The degree of every vertex in this subgraph is:

1. Even
2. Odd
3. Either
4. Integer

Euler Tour!

Background on Eulerian graphs

Definition

An **Euler tour** of an undirected multigraph $G = (V, E)$ is a closed walk that visits each edge exactly once. A graph is **Eulerian** if it has an Euler tour.

Background on Eulerian graphs

Definition

An **Euler tour** of an undirected multigraph $G = (V, E)$ is a closed walk that visits each edge exactly once. A graph is **Eulerian** if it has an Euler tour.

Theorem (Euler)

An undirected multigraph $G = (V, E)$ is Eulerian iff G is connected and every vertex degree is even.

Background on Eulerian graphs

Definition

An **Euler tour** of an undirected multigraph $G = (V, E)$ is a closed walk that visits each edge exactly once. A graph is **Eulerian** if it has an Euler tour.

Theorem (Euler)

An undirected multigraph $G = (V, E)$ is Eulerian iff G is connected and every vertex degree is even.

Theorem

A directed multigraph $G = (V, E)$ is Eulerian iff G is weakly connected and for each vertex v, $indeg(v) = outdeg(v)$.

Improved approximation for Metric-TSP

How can we improve the MST-heuristic?

Observation: Finding optimum TSP tour in G is same as finding minimum cost Eulerian subgraph of G (allowing duplicate copies of edges).

Improved approximation for Metric-TSP

How can we improve the MST-heuristic?

Observation: Finding optimum TSP tour in G is same as finding minimum cost Eulerian subgraph of G (allowing duplicate copies of edges).

Christofides-Heuristic$(G = (V, E), c)$

```
Compute a minimum spanning tree (MST) T in G
Add edges to T to make Eulerian graph H
An Eulerian tour of H gives a tour of G
Obtain Hamiltonian cycle by shortcutting the tour
```

How do we add edges to make T Eulerian?

Christofides Heuristic: 3/2 approximation

Christofides-Heuristic$(G = (V, E), c)$

`Compute a minimum spanning tree (MST)` T `in` G

Christofides Heuristic: 3/2 approximation

Christofides-Heuristic$(G = (V, E), c)$

```
    Compute a minimum spanning tree (MST) T in G
    Let S be vertices of odd degree in T (Note: |S| is even)
```

Christofides Heuristic: $3/2$ approximation

Christofides-Heuristic$(G = (V, E), c)$

```
Compute a minimum spanning tree (MST) T in G
Let S be vertices of odd degree in T (Note:  |S| is even)
Find a minimum cost matching M on S in G
```

Christofides Heuristic: 3/2 approximation

Christofides-Heuristic$(G = (V, E), c)$

```
Compute a minimum spanning tree (MST) T in G
Let S be vertices of odd degree in T (Note: |S| is even)
Find a minimum cost matching M on S in G
Add M to T to obtain Eulerian graph H
```

Christofides Heuristic: 3/2 approximation

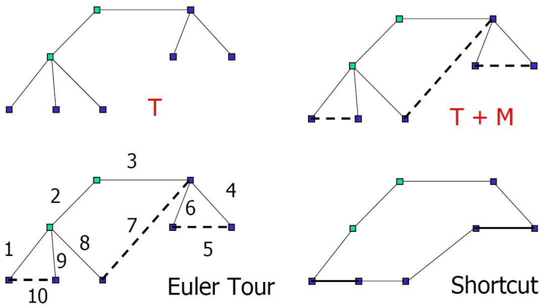
Christofides-Heuristic$(G = (V, E), c)$

```
Compute a minimum spanning tree (MST) T in G
Let S be vertices of odd degree in T (Note: |S| is even)
Find a minimum cost matching M on S in G
Add M to T to obtain Eulerian graph H
An Eulerian tour of H gives a tour of G
Obtain Hamiltonian cycle by shortcutting the tour
```

Christofides Heuristic: 3/2 approximation

Christofides-Heuristic$(G = (V, E), c)$

Compute a minimum spanning tree (MST) T in G
Let S be vertices of odd degree in T (Note: $|S|$ is even)
Find a minimum cost matching M on S in G
Add M to T to obtain Eulerian graph H
An Eulerian tour of H gives a tour of G
Obtain Hamiltonian cycle by shortcutting the tour



Analysis of Christofides Heuristic

Main lemma:

Lemma

$c(M) \leq OPT/2$.

Analysis of Christofides Heuristic

Main lemma:

Lemma

$c(M) \leq OPT/2$.

Assuming lemma:

Theorem

Christofides heuristic returns a tour of cost at most $3OPT/2$.

Proof.

$c(H) = c(T) + c(M) \leq OPT + OPT/2 \leq 3OPT/2$. Cost of tour is at most cost of H. □

Analysis of Christofides Heuristic

Lemma

Suppse $G = (V, E)$ *is a metric and* $S \subset V$ *be a subset of vertices. Then there is a TSP tour in* $G[S]$ *(the graph induced on* S*) of cost at most* OPT.

Analysis of Christofides Heuristic

Lemma

Suppse $G = (V, E)$ *is a metric and* $S \subset V$ *be a subset of vertices. Then there is a TSP tour in* $G[S]$ *(the graph induced on* S*) of cost at most* OPT*.*

Proof.

Analysis of Christofides Heuristic

Lemma

Suppse $G = (V, E)$ is a metric and $S \subset V$ be a subset of vertices. Then there is a TSP tour in $G[S]$ (the graph induced on S) of cost at most OPT.

Proof.

Let $C = v_1, v_2, \ldots, v_n, v_1$ be an optimum tour of cost OPT in G and let $S = \{v_{i_1}, v_{i_2}, \ldots, v_{i_k}\}$ where, without loss of generality $i_1 < i_2 \ldots < i_k$. Then consider the tour $C' = v_{i_1}, v_{i_2}, \ldots, v_{i_k}, v_{i_1}$ in $G[S]$. The cost of this tour is at most cost of C by shortcutting. □

Proof of lemma for Christofides heuristic

Lemma

$c(M) \leq OPT/2$.

Recall that M is a matching on S the set of odd degree nodes in T. Recall that $|S|$ is even.

Proof.

Proof of lemma for Christofides heuristic

Lemma

$c(M) \leq OPT/2$.

Recall that M is a matching on S the set of odd degree nodes in T. Recall that $|S|$ is even.

Proof.

From previous lemma, there is tour of cost OPT for S in $G[S]$. Wlog let this tour be $v_1, v_2, \ldots, v_{2k}, v_1$ where $S = \{v_1, v_2, \ldots, v_{2k}\}$. Consider two matchings M_a and M_b where $M_a = \{(v_1, v_2), (v_3, v_4), \ldots, (v_{2k-1}, v_{2k})$ and $M_b = \{(v_2, v_3), (v_4, v_5), \ldots, (v_{2k}, v_1)$.
$M_a \cup M_b$ is set of edges of tour so $c(M_a) + c(M_b) \leq OPT$ and hence one of them has cost less than $OPT/2$. □

Other comments

Christofides heuristic has not been improved since 1976!
Major open problem in approximation algorithms.

For points in any fixed dimension d there is a polynomial-time approximation scheme. For any fixed $\epsilon > 0$ a tour of cost $(1 + \epsilon)OPT$ can be computed in polynomial time. [Arora 1996, Mitchell 1996].

Excellent practical code exists for solving large scale instances of TSP that arise in several applications. See Concorde TSP Solver by Applegate, Bixby, Chvatal, Cook.

Directed Graphs and Asymmetric TSP (ATSP)

Question: What about directed graphs?

Equivalent of Metric-TSP is Asymmetric-TSP (ATSP)

- Input is a complete directed graph $G = (V, E)$ with edge costs $c : E \rightarrow \mathbb{R}_+$.
- Edge costs are not necessarily symmetric. That is $c(u, v)$ can be different from $c(v, u)$
- Edge costs satisfy assymetric triangle inequality: $c(u, w) \leq c(u, v) + c(v, w)$ for all $u, v, w \in V$.

Directed Graphs and Asymmetric TSP (ATSP)

Question: What about directed graphs?

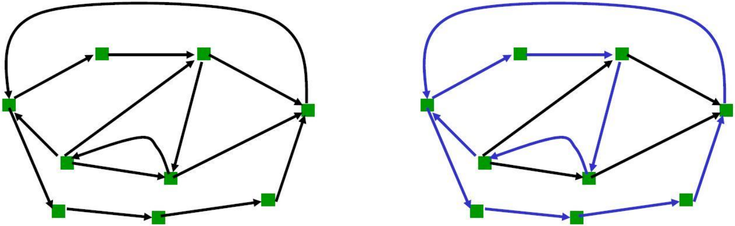
Equivalent of Metric-TSP is Asymmetric-TSP (ATSP)

- Input is a complete directed graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$.
- Edge costs are not necessarily symmetric. That is $c(u, v)$ can be different from $c(v, u)$
- Edge costs satisfy assymetric triangle inequality: $c(u, w) \leq c(u, v) + c(v, w)$ for all $u, v, w \in V$.

Alternate interpretation: given directed graph $G = (V, E)$ find a closed walk that visits all vertices (can visit a vertex more than once).

ATSP

Alternate interpretation: given directed graph $G = (V, E)$ find a closed walk that visits all vertices (can visit a vertex more than once).



Same as finding a minimum cost connected Eulerian subgraph of G.

Approximation for ATSP

Harder than Metric-TSP

- Simple $\log_2 n$ approximation from 1980.
- Improved to $O(\log n / \log \log n)$-approximation in 2010.
- Further improved to $O((\log \log n)^c)$-approximation in 2015.
- Finally to c-approximation in 2018, where $c = 5500$!

Believed that the constant should be much smaller. Lower bound is 2 for LP relaxations.

The O(log n) Approximation

Recall that a cycle cover is a collection of node disjoint cycles that contain all nodes.

The O(log n) Approximation

Recall that a cycle cover is a collection of node disjoint cycles that contain all nodes.

Question: How to find a minimum cost cycle cover?

The O(log n) Approximation

Recall that a cycle cover is a collection of node disjoint cycles that contain all nodes.

Question: How to find a minimum cost cycle cover?

Ans: Reduces to minimum cost bipartite matching!

The $O(\log n)$ Approximation

Recall that a cycle cover is a collection of node disjoint cycles that contain all nodes.

The O(log n) Approximation

Recall that a cycle cover is a collection of node disjoint cycles that contain all nodes.

CycleShrinkingAlgorithm($G(V, A), c : A \to \mathcal{R}^+$):
    If $|V| = 1$ output the trivial cycle consisting of V
    Find a *minimum cost cycle cover* with cycles $C_1, \ldots, C_k$

The O(log n) Approximation

Recall that a cycle cover is a collection of node disjoint cycles that contain all nodes.

CycleShrinkingAlgorithm($G(V, A), c : A \to \mathcal{R}^+$):
    If $|V| = 1$ output the trivial cycle consisting of V
    Find a *minimum cost cycle cover* with cycles $C_1, \ldots, C_k$
    From each C_i pick an arbitrary proxy node v_i
    Let $S = \{v_1, v_2, \ldots, v_k\}$

The O(log n) Approximation

Recall that a cycle cover is a collection of node disjoint cycles that contain all nodes.

CycleShrinkingAlgorithm($G(V, A), c : A \to \mathcal{R}^+$):
    If $|V| = 1$ output the trivial cycle consisting of V
    Find a *minimum cost cycle cover* with cycles $C_1, \ldots, C_k$
    From each C_i pick an arbitrary proxy node v_i
    Let $S = \{v_1, v_2, \ldots, v_k\}$
    Recursively solve problem on $G[S]$ to obtain a solution C
    $C' = C \cup C_1 \cup C_2 \ldots C_k$ is a Eulerian graph.

The O(log n) Approximation

Recall that a cycle cover is a collection of node disjoint cycles that contain all nodes.

CycleShrinkingAlgorithm($G(V, A), c : A \to \mathcal{R}^+$):
- If $|V| = 1$ output the trivial cycle consisting of V
- Find a *minimum cost cycle cover* with cycles $C_1, \ldots, C_k$
- From each C_i pick an arbitrary proxy node v_i
- Let $S = \{v_1, v_2, \ldots, v_k\}$
- Recursively solve problem on $G[S]$ to obtain a solution C
- $C' = C \cup C_1 \cup C_2 \ldots C_k$ is a Eulerian graph.
- Shortcut C' to obtain a cycle on V and output C'.

Illustration and Analysis

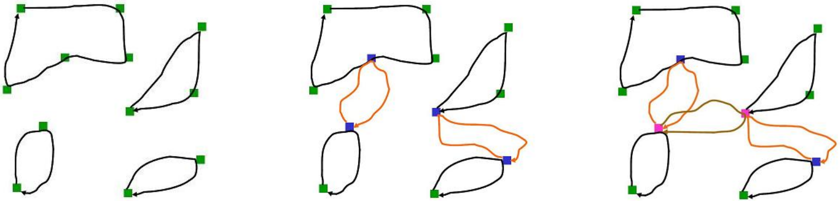
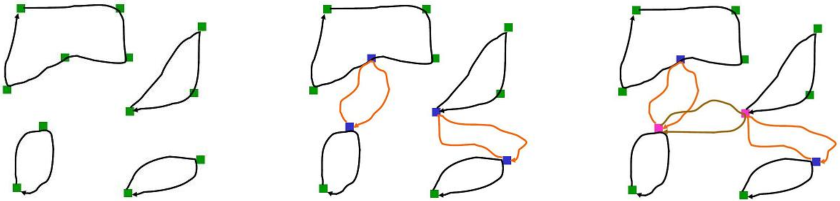


Illustration and Analysis



Analysis: Lemmas

Lemma

Cost of a minimum cost cycle cover *is at most* OPT.

Analysis: Lemmas

Lemma

Cost of a minimum cost cycle cover *is at most* OPT.

Lemma (Proved before)

Suppse $G = (V, E)$ *is a directed graph with edge costs that satisfies asymmetric triangle inequality and* $S \subset V$ *be a subset of vertices. Then there is a TSP tour in* $G[S]$ *(the graph induced on* S*) of cost at most* OPT.

Analysis: Lemmas

Lemma

Cost of a minimum cost cycle cover *is at most OPT.*

Lemma (Proved before)

Suppse $G = (V, E)$ is a directed graph with edge costs that satisfies asymmetric triangle inequality and $S \subset V$ be a subset of vertices. Then there is a TSP tour in $G[S]$ (the graph induced on S) of cost at most OPT.

Lemma

The number of vertices shrinks by half in each iteration and hence total of at most $\lceil \log n \rceil$ cycle covers.

Hence total cost of all cycle covers is at most $\lceil \log n \rceil \cdot OPT$.