Lecture 19

The An-Kleinberg-Shmoys Algorithm for the Traveling Salesman Path Problem*

19.1 The TSP and TSPP Problems

We will assume throughout that we are given a complete undirected graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}^{\geq 0}$ and that these edge costs satisfy the Triangle Inequality ($c(u, w) \leq c(u, v) + c(v, w)$ for all $u, v, w \in V$). The shortest path costs in this graph are then a metric. Recall that a Hamiltonian cycle is a cycle than includes each vertex exactly once and a Hamiltonian path is a path that visits each vertex exactly once.

Now, we define the traveling salesman problem and a couple of its variants

Traveling Salesman Problem (TSP) Find the cheapest Hamiltonian cycle.

Traveling Salesman Path Problem (TSPP) Find the cheapest Hamiltonian path. We can assume without loss of generality that we are given a pair of vertices s and t that we want our path to begin and end at. We can use an algorithm for this case in the general case by running it on all n^2 pairs of vertices.

Asymmetric Traveling Salesman Problem (ATSP) For a complete undirected graph that may have different costs on (u, v) and (v, u) edges, find the cheapest Hamiltonian cycle.

19.2 Previous Results

Previous results for all three problems are summarized in this table:

*Lecturer: Ryan O'Donnell. Scribe: David Witmer.

	TSP	TSPP	ATSP
Best approx factor	3/2 [Chr76]	5/3 [Hoo91]	$\Theta(\log n)$ [FKR95]
LP gap	$\geq 4/3 - o(1)$ [HK70]	$\geq 3/2 - o(1)$ [HK70]	$\geq 4/3 - o(1)$ [HK70]
			$\geq 2 - o(1)$ [CGK06]
LP gap conjecture	$\leq 4/3$	$\leq 3/2$	?

There have also been several recent breakthroughs:

- An $O(\log n / \log\log n)$-approximation algorithm for ATSP [AGM+10]
- A $\leq 3/2 - \epsilon_0$-approximation algorithm (ϵ_0 constant) for graphic TSP, which takes an input graph with unit-length edges and uses as a metric the shortest path metric on the graph [OGSS11]
- A 1.461-approximation algorithm for graphic TSP and a 1.586-approximation algorithm for graphic TSPP [MS11]
- Mucha improved this analysis to get a 1.444-approximation algorithm for graphic TSP and a 1.583 approximation algorithm for graphic TSPP [Muc11]
- There has also been work on the cubic TSP problem, which is TSP on graphs in which all vertices have degree 3 [GLS05], [BSvdSS11]
- A $\frac{1+\sqrt{5}}{2}$-approximation algorithm for TSPP [AKS11]

We will cover the AKS algorithm. We will not give a $\frac{1+\sqrt{5}}{2}$-approximation, but will show a 5/3 approximation algorithm this class and a 1.6583-approximation next class using the major ideas of the paper. These two lectures will cover sections 3.1 through 3.3 of [AKS11].

19.3 A 2-Approximation for TSPP

We will start by giving a 2-approximation for TSPP:

Compute a minimum spanning tree (MST). Note that its cost must be less than or equal to Opt_{TSPP} because any Hamiltonian path is also a spanning tree. We can then walk from s to t on the MST visiting all vertices using each edge at most twice. This gives us a 2-approximation.

To deal with duplicate edges, we can "shortcut" to the next vertex in the MST walk that we haven't yet visited. This will only improve our cost by our assumption that the shortest-path distances are a metric.

It remains to argue that we visit each edge in this path at most once. Consider adding a dummy edge with 0 cost from s to t. Then double each edge except for the dummy edge and the edges on the s-t path in the MST, replacing each with two identical copies. Then each vertex has even degree and the graph is Eulerian. In any Eulerian graph, there exists an Eulerian tour, which is a cycle containing each edge exactly once. Dropping the dummy s-t edge from the cycle gives an s-t path.

To get a better approximation algorithm, we would like to find a cheaper way to get a graph of even degere. We have the following idea from [Hoo91]: Take the MST as before. We want s and t to have odd degree and all other vertices to have even degree. We need to "fix" the parity of any odd-degree vertex that is not s or t. Since the sum of the degrees in any graph is even, we must have an even number of of vertices that need to fixed. See Figure 19.1 for an example. We can therefore add a minimum cost matching on these vertices to fix them and then show that this matching doesn't cost too much. Christofides gave a 3/2-approximation algorithm for TSP by showing that the matching cost is less than or equal to $\frac{1}{2}\text{Opt}_{\text{TSP}}$ [Chr76]. Hoogeveen gave a 5/3-approximation algorithm for TSPP by showing that the matching cost is less than or equal to $\frac{2}{3}\text{Opt}_{\text{TSPP}}$ [Hoo91].

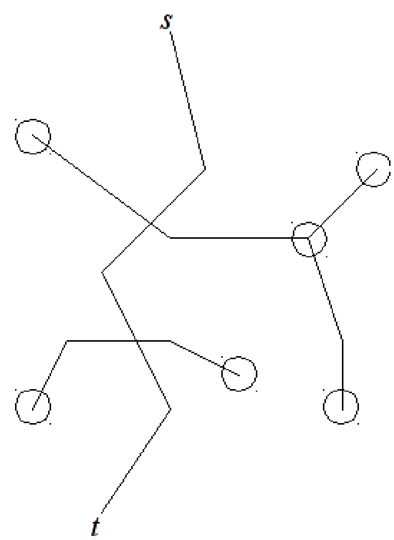

Figure 19.1: A tree with an even number of odd parity "bad" vertices

19.4 LP Relaxations for TSP and TSPP

19.4.1 Solving the LPs

In 1970, Held and Karp gave an LP relaxation for TSP [HK70]:

$$\begin{aligned} \min c(x) := \sum_{e \in E} c_e x_e & \\ \text{s.t. } x(\partial S) \geq 2 & \quad \forall\ S \subsetneq V, |S| > 1 \\ x(\partial S) = 2 & \quad \forall\ S \subsetneq V, |S| = 1 \\ x_e \geq 0 & \quad \forall e \in E \end{aligned}$$

Recall that $\partial S = E(S, \bar{S})$ is the set of edges with one endpoint in S and one endpoint in $\bar{S}$ and $x(F) = \sum_{e \in F} x_e$.

Note that this LP has exponential size. We can still solve it in polynomial time using the ellipsoid algorithm if we have a separation oracle. Thinking of the x_e's as edge capacities, we can find the min cut in polynomial time. If the min cut is ≥ 2, the $x(\partial S) \geq 2$ condition must hold for all valid S. The rest of the constraints are easy to check in polynomial time.

Next, we give an LP relaxation for TSPP:

$$\begin{aligned} &\min c(x) \\ \text{s.t. } & x(\partial S) \geq 1 \quad \text{for separating cuts, i.e. } |S \cap \{s,t\}| = 1, \text{ with } |S| > 1 \\ & x(\partial S) \geq 2 \quad \text{for non-separating cuts with } |S| > 1 \\ & x(\partial S) = 2 \quad \text{for cuts with } |S| = 1 \\ & x_e \geq 0 \quad \forall e \in E \end{aligned}$$

Note that

$$x(E) = \frac{2(n-2)+2}{2} = n-1,$$

ensuring that any integral solution is a Hamiltonian path. Also, this polytope is contained within the MST polytope.

Once again, we need a separation oracle to solve this LP. We take the x_e's to be edge capacities as before and add an additional s-t edge with capacity 1. If the min cut ≥ 2, the two $\geq$ constraints are satisfied and the other constraints can be checked as before in polynomial time.

19.4.2 Integrality Gaps

We will now show graphs demonstrating integrality gaps of 3/2 and 4/3 for the TSPP and TSP LP relaxations, respectively.

For TSPP, consider Figure 19.2. Any s-t path containing all of the edges is going to have about 3ℓ edges in it. However, we can find a feasible solution to the LP with 1/2 on the edges out of s and t and the two vertical edges and 1 on all other edges to see that LPOpt is about 2ℓ. This shows that the integrality gap is at least 3/2.

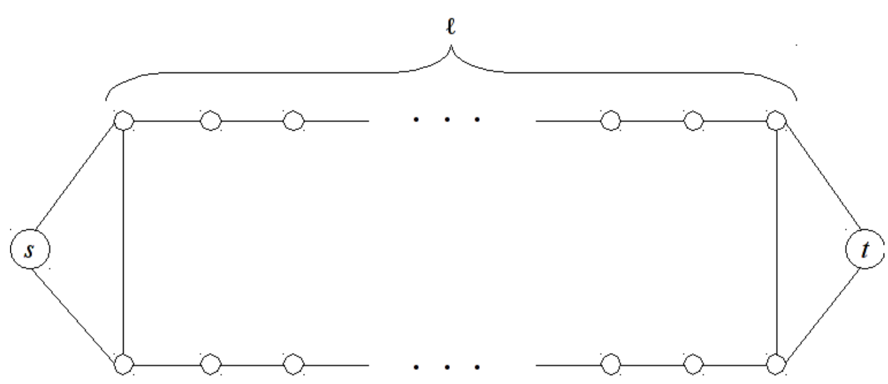

Figure 19.2: A graph showing the 3/2 integrality gap for TSPP

A similar graph with an extra row of vertices demostrates the 4/3 integrality gap for TSP.

19.4.3 The Spanning Tree Polytope

Consider the spanning tree polytope:

$$\begin{aligned} x(E) &= n-1 \\ x(\partial S) &\geq |S|-1 \quad \forall\, S \subsetneq V, |S|>1 \\ x(\partial S) &\geq 1 \quad \forall\, S \subsetneq V, |S|=1 \\ x_e &\geq 0 \quad \forall e \in E \end{aligned}$$

The TSPP polytope is contained within the intersection of the spanning tree polytope and the set $\{x \mid x(\partial\{s\}) = x(\partial\{t\}) = 1\}$. Since the spanning tree polytope is integral and this polytope is the convex hull of these integral solutions, we can write any TSPP LP solution as a convex combination of spanning trees in which s and t are leaves. That is, for non-negative λ_i that sum up to 1

$$x = \sum_{i \leq \binom{n}{2}} \lambda_i \mathbf{1}_{A_i}$$

where A_i is a spanning tree in which s and t are leaves. Furthermore, we can find this linear combination efficiently.

19.5 The AKS Algorithm

Now we state the AKS algorithm.

1. Solve the TSPP LP relaxation to get a solution x^*.
2. Write x^* as a convex combination of spanning trees A_i that have s and t as leaves to get $x^* = \sum_{i \leq \binom{n}{2}} \lambda_i \mathbf{1}_{A_i}$.
3. Pick a spanning tree A at random from this distribution (choose A_i with probability λ_i).
4. Let $T = T_A$ be the set of vertices in A whose degree parity needs to be fixed. As described above, $|T|$ is even.
5. Take M to be the minimum cost matching on T.
6. Then $A \cup M$ has an Eulerian path from s to t. Shortcut to avoid taking the same edge twice and return the resulting path.

This algorithm can be derandomized by trying all $\binom{n}{2}$ of the A_i's.

We want to show that the expected cost of this algorithm is strictly less than 5/3. Note that

$$\mathbf{E}[\text{cost}(A)] = \sum_i \lambda_i \text{cost}(A_i) = c(x^*) \leq \text{Opt}.$$

In the next lecture, we will get our desired result by showing that $\mathbf{E}_A[\text{cost}(M)] \leq .6853\bar{3}c(x^*)$, giving an $1.6583\bar{3}$-approximation algorithm. In this lecture, we will show that $\mathbf{E}_A[\text{cost}(M)] \leq \frac{2}{3}c(x^*)$ to get a 5/3-approximation algorithm.

19.5.1 T-Joins

First, we need to define some more terminology.

Definition 19.1. Given a subset of vertices T, a T-join is a subgraph such that all the vertices of T have odd degree and all other vertices have even degree.

A T-join consists of $|T|/2$ edge-disjoint paths connecting pairs on T plus cycles. In the metric case, the cheapest T-join is a matching.

We can find the cheapest or most expensive T-join in polynomial time because the T-join polytope is integral:

$$\begin{aligned} y(\partial S) &\geq 1 \quad \text{for all odd-}T\text{-cuts, i.e. cuts such that } |S \cap T| \text{ is odd} \\ y_e &\geq 0 \quad \forall e \in E \\ y_e &\leq 1 \quad \forall e \in E \end{aligned}$$

Theorem 19.2. [EJ73] *This polytope is integral.*

We call any y satisfying these conditions a fractional T-join. We can remove the $y_e \geq 1$ constraint. Any y satisfying the remaining constraints is called a fractional T-join dominator.

Given these definitions and facts, we can then write

$$\begin{aligned} \text{cost of cheapest matching on } T &= \text{cost of cheapest } T\text{-join} \\ &= \text{cost of cheapest fractional } T\text{-join} \\ &\leq \text{cost of any fractional } T\text{-join} \\ &\leq \text{cost of any fractional } T\text{-join dominator.} \end{aligned}$$

19.5.2 A 5/3-Approximation Algorithm for TSPP

In the end, our goal is to give a mapping from (x^*, A) to a fractional T-join dominator y such that $\mathbf{E}_A[c(y)] \leq .6583\bar{3}c(x^*)$. By the above discussion, $\mathbf{E}_A[\text{cost}(M)] \leq \mathbf{E}_A[c(y)]$, so this will give us a $1.6583\bar{3}$-approximation algorithm as desired. For today, we will show that $\mathbf{E}_A[c(y)] \leq \frac{2}{3}c(x^*)$.

We'll try a couple of ideas:

Idea 0 Set $y = x^*$. x^* is a fractional T-join dominator by the constraints of the TSPP LP relaxation. However, this would only give us a 2-approximation. What happens if we choose $y = \frac{1}{2}x^*$? This will not work since the $y(\partial S) \geq 1$ constraint could be violated for separating cuts. Note that for TSPs, this would work and would give a 3/2-approximation.

What about $y = \mathbf{1}_A$? This will work, since A is a tree.

We need the following proposition to analyze our next idea:

Proposition 19.3. *If S is an s-t-separating odd-T-cut, then $\mathbf{1}_A(\partial S) \geq 2$.*

Proof. Recall that s and t are leaves in A. S can be partitioned into vertices in $S \cap T$, a set containing either only s or only t, and a set containing no vertices of T. Because the cut is an odd-T-cut, there will be an odd number of vertices in the first set. Each will have odd degree becasuse they are al in T. The sum of the degrees of vertices in the first set is therefore odd. Since s and t are leaves, the degree of the second set i 1. Since all vertices in the third set are not in T, the sum of their degrees must be even. So, we get that the sum of the degrees of vertices in S is an odd number plus 1 plus an even number, which must be even. This implies that $|\partial S|$ is even. A is a spanning tree, so it is connected. It must then be the case that $|\partial S| \geq 2$. $\square$

An example cut is shown in Figure 19.3.

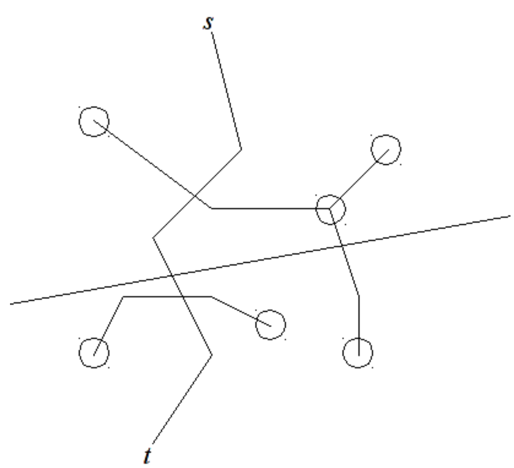

Figure 19.3: A tree with the vertices of T circled and a cut satisfying the conditions of the proposition shown

Idea 1 Set $y = \frac{1}{3}x^* + \frac{1}{3}\mathbf{1}_A$. Then we would have that $E_A[\text{cost}(y)] = \frac{2}{3}c(x^*)$ as desired. We just need to show that y is a fractional T-join dominator. Consider S non-separating. Then

$$y(\partial S) = \frac{1}{3}x^*(\partial S) + \frac{1}{3}\mathbf{1}_A(\partial S) \geq \frac{2}{3} + \frac{1}{3} = 1$$

by the LP constraints and the fact that A is a tree. If, on the other hand, S is separating and odd, we have that

$$y(\partial S) \geq \frac{1}{3} + \frac{2}{3} = 1$$

by the proposition and the LP constraints.

Therefore, we have a 5/3-approximation for TSPP.

Bibliography

[AGM$^+$10] Arash Asadpour, Michel X. Goemans, Aleksander Madry, Shayan Oveis Gharan, and Amin Saberi. An $o(\log n/\log\log n)$-approximation algorithm for the asymmetric traveling salesman problem. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pages 379 – 189, 2010. 19.2

[AKS11] Hyung-Chan An, Robert Kleinberg, and David B. Shmoys. Improving christofides' algorithm for the s-t path tsp. *CoRR*, abs/1110.4604, 2011. 19.2

[BSvdSS11] Sylvia Boyd, Rene Sitters, Suzanne van der Ster, and Leen Stougie. Tsp on cubic and sub-cubic graphs. In *Proceedings of the 15th Conference on Integer Programming and Combinatorial Optimization*, IPCO '11, 2011. 19.2

[CGK06] Moses Charikar, Michel X. Goemans, and Howard Karloff. On the integrality ratio for the asymmetric traveling salesman problem. *Mathematics of Operations Research*, 31:245 – 252, 2006. 19.2

[Chr76] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, CMU Graduate School of Industrial Administration, 1976. 19.2, 19.3

[EJ73] Jack Edmonds and Ellis L. Johnson. Matching, euler tours and the chinese postman. *Mathematical Programming*, 5:88 – 124, 1973. 19.2

[FKR95] Alan Frieze, Richard M. Karp, and Bruce Reed. When is the assignment bound tight for the asymmetric traveling-salesman problem? *SIAM J. Comput.*, 24:484 – 493, June 1995. 19.2

[GLS05] David Gamarnik, Moshe Lewenstein, and Maxim Sviridenko. An improved upper bound for the tsp in cubic 3-edge-connected graphs. *Operations Research Letters*, 33:467 – 474, 2005. 19.2

[HK70] Michael Held and Richard M. Karp. The traveling-salesman problem and minimum spanning trees. *Operations Research*, 18:1138 – 1162, 1970. 19.2, 19.4.1

[Hoo91] J.A. Hoogeveen. Analysis of christofides' heuristic: Some paths are more difficult than cycles. *Operations Research Letters*, 10(5):291 – 295, 1991. 19.2, 19.3

[MS11] Tobias Mömke and Ola Svensson. Approximating graphic tsp by matchings. *CoRR*, abs/1104.3090, 2011. 19.2

[Muc11] Marcin Mucha. Improved analysis for graphic tsp approximation via matchings. *CoRR*, abs/1108.1130, 2011. 19.2

[OGSS11] Shayan Oveis Gharan, Amin Saberi, and Mohit Singh. A randomized rounding approach to the traveling salesman problem. In *52nd Annual IEEE Symposium on Foundations of Computer Science*, FOCS '11, 2011. 19.2